

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Fakin

**Prototip sistema za oceno kakovosti turističnih točk na  
podlagi obstoječih ocen iz spletnih virov**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2016



UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Fakin

**Prototip sistema za oceno kakovosti turističnih točk na  
podlagi obstoječih ocen iz spletnih virov**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana, 2016



Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Prototip sistema za oceno kakovosti turističnih točk na podlagi obstoječih ocen iz spletnih virov

Tematika naloge:

Na spletu se pojavlja vrsta strani, ki so namenjene ocenjevanju različnih turističnih točk, kot npr. prenočišča, restavracije, turistične zanimivosti ipd. Primerjava ocen posameznih turističnih točk na različnih straneh včasih pokaže tudi večje razlike. V okviru naloge najprej predstavite področje ter morebitne sorodne rešitve, nato pa izdelajte prototip informacijskega sistema, ki bo omogočil enostavno primerjavo izbranih turističnih točk glede na njihovo oceno na različnih spletnih straneh. Sistem naj med drugim omogoča beleženje zgodovine ocen in s tem tudi zaznavanje trenda. Izračun skupne ocene naj temelji na obstoječih ocenah, pri čemer kot utež upoštevajte tudi število ocenjevalcev. Sistem naj obsega tako zaledni kot čelni del, pri čemer poskrbite, da bo čelni del deloval na različnih (tudi mobilnih) napravah. Prototip preizkusite na primerih več različnih turističnih točk. Delovanje prototipa kritično ovrednotite ter podajte smernice za nadaljnje delo.





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matej Fakin sem avtor diplomskega dela z naslovom:

*Prototip sistema za oceno kakovosti turističnih točk na podlagi obstoječih ocen iz spletnih virov*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 16. septembra 2016

Podpis avtorja:



*Zahvaljujem se mentorju doc. dr. Damjanu Vavpotiču za pomoč in koristne nasvete pri izdelavi diplomskega dela. Prav tako gre zahvala tudi družini, ki me je podpirala skozi celoten študij ter pri izdelavi diplomskega dela.*



# Kazalo

**Povzetek**

**Abstract**

<b>Poglavje 1</b>	<b>Uvod .....</b>	<b>1</b>
<b>Poglavje 2</b>	<b>Predstavitev področja .....</b>	<b>3</b>
<b>Poglavje 3</b>	<b>Razvojna orodja in tehnologije .....</b>	<b>5</b>
3.1	Uvod.....	5
3.2	Razvojna orodja .....	5
3.2.1	Sublime Text .....	5
3.2.2	PhpMyAdmin .....	5
3.2.3	Cpanel.....	6
3.3	Tehnologije .....	6
3.3.1	PHP .....	6
3.3.2	MySql .....	7
3.3.3	HTML in CSS.....	7
3.3.4	JavaScript in jQuery .....	7
3.3.5	AJAX.....	8
3.3.6	JSON.....	8
3.3.7	Bootstrap.....	8
<b>Poglavje 4</b>	<b>Spletno luščenje podatkov .....</b>	<b>11</b>
4.1	Uvod.....	11
4.2	Metode in knjižnice .....	12
4.2.1	cURL .....	12
4.2.2	Regularni izrazi.....	12
4.2.3	XPath in DOMXPath.....	13

4.2.4	PHP Simple HTML DOM Parser .....	13
4.3	Legalnost luščenja .....	14
<b>Poglavje 5</b>	<b>Razvoj sistema .....</b>	<b>15</b>
5.1	Načrt sistema .....	15
5.1.1	Uvod.....	15
5.1.2	Primeri uporabe.....	15
5.1.3	Podatkovni model .....	17
5.1.4	Implementacijski pogled .....	19
5.1.5	Postopek računanja globalne ocene .....	20
5.2	Pridobivanje podatkov.....	21
5.2.1	Google API .....	21
5.2.2	Foursquare API .....	23
5.2.3	Tripadvisor.....	25
5.2.4	Booking.....	26
5.2.5	Hostelworld.....	28
5.3	Pregled spletne aplikacije.....	29
5.3.1	Nadzorna plošča.....	29
5.3.2	Osrednja aplikacija.....	34
5.4	Samodejno posodabljanje ocen – sistemski klic Cron .....	40
<b>Poglavje 6</b>	<b>Sklepne ugotovitve .....</b>	<b>43</b>

## Seznam uporabljenih kratic

Kratica	Angleško	Slovensko
<b>AJAX</b>	Asynchronous JavaScript and XML	Asinhroni Javascript in XML
<b>API</b>	Application Programming Interface	Programski vmesnik
<b>CSS</b>	Cascading Style Sheets	Kaskadne stilske predloge
<b>DOM</b>	Document Object Model	Objektni model dokumenta
<b>HTML</b>	HyperText Markup Language	Označevalni jezik
<b>HTTP</b>	Hypertext Transfer Protocol	Protokol za prenos nadbesedil
<b>JSON</b>	JavaScript Object Notation	Notacija objektov JavaScript
<b>PHP</b>	Hypertext Preprocessor	Skriptni programski jezik
<b>REST</b>	Representational State Transfer	Arhitektura za izmenjavo podatkov
<b>SOAP</b>	Simple Object Access Protocol	Protokol za izmenjavo podatkov
<b>SQL</b>	Structured Query Language	Strukturiran povpraševalni jezik
<b>URL</b>	Uniform Resource Locator	Enolični krajevnik vira
<b>XHTML</b>	Extensible Hypertext Markup Language	Razširljiv označevalni jezik
<b>XML</b>	Extensible Markup Language	Razširljiv označevalni jezik





## Povzetek

**Naslov:** Razvoj prototipa informacijske rešitve za oceno turističnih zanimivosti oz. drugih turističnih točk na podlagi podatkov iz socialnih omrežij

Turistična panoga dosega na spletu v zadnjih letih velik porast. Turiste na spletu zanimajo ocene in komentarji turističnih točk, ki jih nameravajo obiskati in so jih drugi turisti že ocenili ter podali svoje mnenje. Spletnih virov, ki nudijo takšne ocene, je veliko, zato se pri pregledovanju ocen pojavlja težava, da imamo omejen pogled na vsak vir posebej. Da bi to težavo rešili, smo v okviru diplomskega dela razvili prototip sistema, ki omogoča hiter pregled ocen vseh turističnih točk skupaj na enem mestu. Za lažji pregled smo turistične točke še dodatno razdelili po kategorijah. Sistem je razdeljen na dva dela. Prvi del je odzivna spletna aplikacija, ki služi prikazu podatkov o turističnih točkah in krajih. Drugi del pa predstavlja nadzorna plošča, preko katere upravljamo vse podatke, ki se prikazujejo v osrednji aplikaciji. Osrednja aplikacija nam ponuja še iskalnik turističnih točk in krajev ter prikaz turističnih točk na zemljevidu. Poleg tega imamo na voljo še podrobnejši pregled spreminjanja ocen skozi čas. Aplikacijo smo preizkusili na turističnih točkah iz pet večjih slovenskih krajev. Glede na to, da so se pri kar 25 % primerov med ocenami po posameznih virih pojavile večje razlike, se je takšen pregled ocen izkazal kot uporaben in smiseln.

**Ključne besede:** odzivna spletna aplikacija, spletno luščenje podatkov, turizem, pregled ocen turističnih točk.



## **Abstract**

**Title:** Prototype of a system for evaluation of tourist spots quality based on existing evaluations from web sources

In recent years, the tourism industry is significantly growing on the internet. On the internet, tourists are interested in ratings and comments about tourist spots, which they want to visit and which were already rated and commented upon by other tourists. There are lots of online sources, which provide such ratings, therefore the overview of these ratings is problematic, since we have limited access to each individual source. In order to solve this problem within the thesis, we developed a prototype of a system, which enables a fast overview of ratings of all tourist spots in one place. To further simplify the overview, we categorized the tourist spots. The system is divided into two parts. The first part is a responsive web application for the presentation of data about tourist spots and sites. The second part is a control panel, through which we manage all the data, which are being shown in the main application. The main application also features a search engine to find locations and sites and a display of tourist spots on a map. In addition it offers a more detailed overview of changes of ratings over time. We tested the application on tourist spots from five larger Slovenian towns. Considering that in 25 % of cases the ratings from individual sources differed by more than one, such overview of ratings proved to be useful and meaningful.

**Keywords:** responsive web application, online data scaling, tourism, overview of the ratings of tourist points.



## Poglavje 1      Uvod

Dandanes je dostop do spleta mogoč iz veliko naprav, zato nam je na voljo vedno večja količina podatkov iz vseh področij. V zadnjih letih velik porast dosega tudi turistična panoga. Večina rezervacij prenočišč je izvedenih preko spleta. Preden se ljudje odločijo za rezervacijo določenega hotela, obiska restavracije, znamenitosti itd., pogosto najprej preko spleta raziščejo te turistične točke ter si pregledajo razne komentarje in ocene. Šele po temeljitem pregledu se potem odločijo za rezervacijo oz. obisk. Takšni podatki so načeloma zelo relevantni, saj jih najpogosteje napišejo ljudje, ki so dejansko te točke tudi res obiskali. V zadnjem času je zato na voljo ogromno spletnih portalov, kjer lahko pridobimo takšne podatke. Problem pa se pojavi, ker imamo na voljo preveliko količino podatkov in porabimo preveč časa, da si ustvarimo svoje mnenje. Zato smo se odločili raziskati, kako takšne najbolj relevantne podatke združiti v en sistem, ki bi bil sposoben prikazovati podatke izbrane turistične točke iz več spletnih virov. Usmerili smo se na prikaz ocen turističnih točk in kako se te ocene spreminjajo skozi čas.

V diplomskem delu je predstavljen razvoj prototipa odzivne spletne aplikacije (angl. responsive web application) za prikaz skupnih ocen turističnih točk ter krajev iz več izbranih spletnih virov preko programskih vmesnikov oz. s spletnim luščenjem podatkov. Vsi podatki se urejajo preko nadzorne plošče, ki je dostopna le za izbrane prijavljene uporabnike. Ker so turistične točke ročno vnesene, vsebujejo pravilne URL-povezave spletnih virov, ki služijo pridobivanju ocen. S tem rešimo težavo, da bi turistična točka imela URL-povezavo, ki ne bi vsebovala pravih ocen, saj nekateri turistični viri vsebujejo večkratne vnose iste turistične točke. Glavna prednost sistema je možnost pregleda spreminjanja ocen skozi čas. Omogočeno je tudi iskanje turističnih točk in krajev glede na podane parametre in možnost pregleda vseh turističnih točk na enotnem zemljevidu.

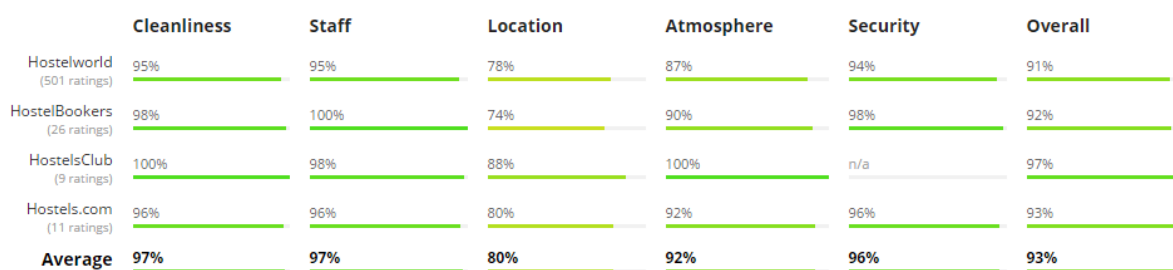
V prvem delu diplomskega dela najprej sledi predstavitev področja. V nadaljevanju so nato predstavljena razvojna orodja, med katerimi najdemo Sublime, PhpMyAdmin ter Cpanel. Nato sledi opis uporabljenih tehnologij, kot so PHP, MySQL, HTML in CSS, JavaScript in jQuery, AJAX, JSON ter Bootstrap. V glavnem delu je predstavljen princip spletnega luščenja podatkov in načinov uporabe. Sledi predstavitev razvoja sistema, ki zavzema

arhitekturo sistema, opis pridobivanja ocen in pregled uporabniškega vmesnika. V zadnjem delu pa je podan sklep ter predstavljene možnosti nadgradnje sistema v prihodnosti.

## Poglavje 2 Predstavitev področja

Prikaz iste vrste podatkov iz več različnih spletnih virov na enem mestu nam je danes na voljo za veliko različnih področij. Splošni primer je primerjava cen izdelkov iz več spletnih trgovin za iskanje najugodnejše cene izdelka. V turistični panogi pa najdemo možnosti, kot so iskanje letalskih povezav, primerjave cen nočitev, iskanje prevozov do letališč itd.

Namen diplomskega dela je bil pridobitev celovitih ocen različnih vrst turističnih točk, na podlagi ocen iz različnih spletnih virov. Ker se te ocene lahko zelo razlikujejo, se nam je prikaz takšnih ocen zdel smiseln in uporaben. Po temeljitem pregledu obstoječih rešitev smo prišli do ugotovitve, da ima takšen pregled ocen iz različnih spletnih virov le spletni portal Hostelz (slika 2.1), pri čemer pa so njegove funkcionalnosti precej omejene. Po preprosti analizi smo ugotovili, da je njegova največja slabost, da vsebuje le ocene o prenočiščih oz. hostlih. Ocene so prikazane kot odstotki izbranih lastnosti, ki se nato po posebnem algoritmu, ki vključuje tudi število ocen, izračunajo v skupno globalno oceno.



Slika 2.1: Prikaz ocen na spletnem portalu Hostelz.

Omenjeno obstoječo rešitev smo uporabili kot zgled pri razvoju lastne rešitve, ki pa smo jo nadgradili z dodatnimi funkcionalnostmi. Naša rešitev se razlikuje po tem, da vsebuje ocene različnih vrst turističnih in ni omejena le na prenočišča. Poleg tega vsebuje pregled zgodovine ocen, kar je naša glavna prednost. Da uporabnik pridobi celovitejši pregled, smo z uporabo ustreznih barv prikazali tudi padanje ali rast ocen v obdobju enega meseca. Za lažje iskanje ocen smo dodatno razdelili turistične točke na več kategorij. To so atrakcije, hrana in pijača, muzeji, nakupovanje, nočno življenje ter prenočišča.





## **Poglavje 3      Razvojna orodja in tehnologije**

### **3.1    Uvod**

Za razvoj prototipa spletne aplikacije je bilo uporabljenih več različnih orodij in tehnologij. Odločili smo se za uporabo takšnih orodij, ki nudijo podporo za izbrane tehnologije in jih uporabljamo vsakodnevno. Tehnologije smo izbrali glede na potrebe, ki jih je morala imeti spletna aplikacija. Najpomembnejše lastnosti, ki so odločale o izbiri, so bile, da jih lahko uporabljamo na našem spletnem gostovanju, da je na voljo veliko dokumentacije in primerov uporabe ter da bodo omogočale izdelavo odzivne spletne aplikacije. Poleg tega pa smo izbrali tudi na podlagi izkušenj, ki jih imamo z razvojem spletnih aplikacij in nam bližjih tehnologij. V nadaljevanju sledi opis le-teh.

### **3.2    Razvojna orodja**

#### **3.2.1 *Sublime Text***

Gre za tekstovni urejevalnik izvorne kode, ki navdušuje s svojo lahkotnostjo, hitrostjo in dovršeno lepim estetskim videzom. Glavne prednosti urejevalnika so hitro iskanje in odpiranje datotek ter možnost preklapljanja med projekti. Poleg tega ponuja več možnosti izbire barvne sheme izvorne kode in ozadja. Zelo uporabna funkcionalnost, ki jo ima, je tudi samo dokončanje (angl. autocomplete) izvorne kode, kar je zelo priročno, saj lahko preko bližnjic ustvarimo izseke določene programske kode brez celotnega pisanja le-te. Vse nastavitve se lahko prilagodijo preko JSON datotek [8]. Velika prednost pa je tudi možnost uporabe dodatnih funkcionalnosti preko vtičnikov, ki jih razvijajo ostali razvijalci in s tem dodatno podpirajo razvoj.

#### **3.2.2 *PhpMyAdmin***

Gre za brezplačno odprtokodno orodje napisano v programskem jeziku PHP, ki ga lahko uporabljamo preko spletnega brskalnika, saj se nahaja na spletnem strežniku in je zato neodvisno od operacijskega sistema. Ponuja nam administracijo podatkovne baze MySQL ter v novejših različicah tudi njegovega naslednika MariaDB. Omogoča večji nabor operacij, med njimi najdemo vse osnovne operacije, kot so dodajanje, urejanje, kopiranje, brisanje

podatkovnih baz, tabel, atributov in njihovih vrednosti. Na voljo imamo tudi naprednejše funkcionalnosti, kot je upravljanje uporabnikov in njihovih pravic, ter celotno kontrolo nad procedurami in triggerji. Glavna prednost pa je zagotovo uporabniški vmesnik, ki je zelo preprost in pregleden za njegovo uporabo. Poleg tega je na voljo ogromno dokumentacije in primerov uporabe, kar lahko zelo olajša delo s podatkovnimi bazami. Nameščen je tudi na večino strežnikov podjetji, ki ponujajo spletna gostovanja in uporabljajo MySQL oz. MariaDB podatkovno bazo [2, 9].

### 3.2.3 Cpanel

Gre za plačljivo spletno orodje namenjeno upravljanju spletnega gostovanja, ki je nameščeno na spletnem strežniku. Orodje nudi poleg upravljanja spletnega gostovanja preko konzole tudi možnost upravljanja preko grafične kontrolne plošče, ki dodatno poenostavi uporabo. Je med najbolj priljubljenimi nadzornimi ploščami, ki služijo upravljanju spletnega gostovanja. S tremi ravnmi arhitekture omogoča preprosto upravljanje celotnega sistema spletnim razvijalcem, administratorjem in prodajalcem spletnih gostovanj. Vključuje uporabo osnovnih funkcionalnosti, kot so upravljanje datotek, vzdrževanje uporabniških FTP-računov in e-pošte, upravljanje domen in podatkovnih baz kot tudi možnost ogleda statistike celotnih parametrov spletnega gostovanja. Omogoča tudi naprednejše funkcionalnosti, kot so varnostne kopije datotek, oddaljen dostop do podatkovne baze, filtriranje e-sporočil proti nezaželeni spletni pošti (angl. spam) in še ogromno ostalih možnosti [10].

## 3.3 Tehnologije

### 3.3.1 PHP

OdpriTokodni programski jezik PHP je bil zasnovan za splet. Uporabljen je na veliko dinamičnih spletnih straneh, saj je zelo priljubljen pri razvijalcih zaradi njegove preprostosti. Uporabljen je tudi pri več priljubljenih CMS-sistemih, kot je npr. Wordpress. Razvijalcem je na voljo veliko dobre dokumentacije in ogromna skupnost ostalih razvijalcev, ki razvijajo uporabna orodja, ki lajšajo proces razvoja v tem programskem jeziku. Primeren je tudi za začetnike, saj nima striktnih pravil o načinih razvoja funkcij in uporabi metod. PHP je zelo toleranten do napak, zato lahko program deluje kljub manjšim napakam. Žal pa lahko ta dinamičnost programskega jezika pri večjih in zahtevnih aplikacijah privede do tega, da je aplikacijo težko vzdrževati, saj lahko te napake postanejo težko izsledljive in popravljive. Z izkušnjami ugotovimo primeren način oblikovanja programske kode za lažje vzdrževanje. Zaradi dinamičnosti pa programski jezik upočasnjuje hitrost delovanja aplikacij. V zadnjih verzijah (PHP 7) so očitne spremembe za izboljšanje hitrosti delovanja in nedoslednosti pri

napakah. Največ je k temu pripomoglo podjetje Facebook, ki v zadnjem času veliko investira v ta programski jezik [5, 11]. V diplomskem delu smo se za njegovo uporabo odločili predvsem zaradi njegovega poznavanja in praktičnih izkušenj pri razvoju spletnih aplikacij.

### **3.3.2 *MySql***

MySql je sistem za upravljanje s podatkovnimi bazami. Je odprtokodna implementacija relacijske podatkovne baze, ki za delo s podatki uporablja programski jezik SQL. Deluje po principu odjemalec – strežnik. Obstaja veliko število odjemalcev in programskih vmesnikov za upravljanje podatkovne baze MySql. Deluje na več operacijskih sistemih in ga podpira večje število programskih jezikov, za katere obstajajo programski vmesniki za povezavo na bazo [12]. Njegova uporaba je zelo priljubljena pri razvijalcih, saj ima poleg dobre dokumentacije še ogromno različnih spletnih strani in ostalega gradiva, s katerim si lahko pomagajo pri razvijanju aplikacij. V aplikaciji ga uporabljamo za shranjevanje vseh podatkov o turističnih točkah, krajih, uporabnikih in globalnih nastavitvah.

### **3.3.3 *HTML in CSS***

Za oblikovanje ter tudi prikaz vsebine statične spletne aplikacije je nujna uporaba HTML, saj brez njega aplikacija ne more delovati. Večinoma se uporablja v povezavi s CSS, ki je oblikovni jezik, s katerim je mogoče oblikovanje stila HTML elementov. Osnove HTML in CSS so zelo preproste, zato večina spletnih razvijalcev najprej začne z njima. Poudariti je potrebno, da to nista programska jezika sama po sebi. Začetnikom ponujata predstavitev konceptov programske sintakse in pravil [13]. Glavna prednost CSS je izogibanje ponavljanju pisanja enakih stilskih lastnosti za vsak element posebej, saj lahko z njim enake oblikovne lastnosti različnih elementov definiramo le enkrat na enem mestu.

### **3.3.4 *JavaScript in jQuery***

Javascript je programski jezik, ki omogoča dodajanje interaktivnosti HTML-elementom z animacijami in vizualnimi učinki. Izumili so ga v podjetju Netscape in je jezik, za katerega lahko rečemo, da je star toliko kot sam splet. Danes je JavaScript zelo upoštevan in uporabljen za razliko od njegove preteklosti. Včasih se je uporabljal za dodajanje razmeroma nekoristnih učinkov, npr. ob premiku miške, ki so bili večinoma odvečni elementi spletne aplikacije. Na spletu je bilo na voljo ogromnih takšnih skript, ki pa niso bile podprte v vseh brskalnikih oz. so delovale na različne načine. Na srečo so takšni dnevi mimo in danes večina sodobnejših brskalnikov omogoča enak prikaz rezultatov napisane JavaScript programske kode. Še vedno je nekaj neskladnosti zaradi nekaterih nestandardiziranih elementov, ki jih vsak brskalnik različno interpretira. Z ustreznim pristopom pa se lahko te težave zaobidejo oz. popravijo. V

zadnjih nekaj letih je JavaScript doživel preporod, saj ga uporablja kar nekaj odmevnih spletnih strani, kot so Google, Yahoo in Flickr, ki ga uporabljajo v veliki meri za ustvarjanje interaktivnih spletnih aplikacij. JavaScript programsko kodo je mnogim razvijalcem kljub preprostim ukazom težko pisati, saj je velikokrat koda zelo dolga, pojavlja pa se že zgoraj opisana neskladnost z brskalniki pri kompleksnejših elementih. Tukaj se pojavi JavaScript knjižnica po imenu jQuery, ki odpravi takšne neskladnosti in omogoča lažje razvijanje. Knjižnica vsebuje skupek zapletene JavaScript kode, združene v krajše metode, ki poenostavljajo njegovo uporabo. Z drugimi besedami jQuery rešuje kompleksnost JavaScript programske kode [3]. V naši aplikaciji uporabljamo različne knjižnice, ki temeljijo na knjižnici jQuery za prikaz grafov in vizualnih učinkov elementov.

### 3.3.5 AJAX

Je skupina medsebojno povezanih spletnih razvojnih tehnik, ki se uporabljajo pri izdelavi dinamičnih spletnih strani. Podatki se s pomočjo jezika JavaScript v ozadju asinhrono pošiljajo strežniku, ki jih obdela in nato posodobi le določen element brez potrebe po ponovnem nalaganju celotne spletne strani. Tehnologije, ki so vključene, so HTML in CSS za predstavitev, DOM za dinamični prikaz in interakcijo s podatki, JSON ali XML za izmenjavo podatkov, XMLHttpRequest objekt za asinhrono komuniciranje in JavaScript, ki povezuje vse našete tehnologije [1].

### 3.3.6 JSON

JSON je odprt standard tekstovnih datotek za izmenjavo podatkov v človeku razumljivi obliki. Izhaja iz programskega jezika JavaScript in je tekstovni format, ki je neodvisen od programskega jezika. Omogoča nam lažje prenašanje podatkov preko različnih programskih jezikov v enotni obliki. Sestavljen je iz parov vrednosti in ključev. Je najpogostejša oblika podatkov, ki se uporablja za asinhrono komunikacijo pri AJAX-zahtevah, saj je zelo preprost in enostavnejši za uporabo od konkurenčnega XML-ja [14, 15]. V aplikaciji ga uporabljamo pri AJAX-zahtevah, ki pridobivajo podatke preko programskega jezika PHP in so nato prikazani preko JavaScript v grafih in tabelah.

### 3.3.7 Bootstrap

Bootstrap je odprtokodno ogrodje (angl. framework), razvito z uporabo HTML, CSS in JavaScript. Sprva sta ga razvila oblikovalec in razvijalec podjetja Twitter in je bilo namenjeno interni uporabi za spodbujanje skladnosti med notranjimi orodji. Nato je orodje postalo odprtokodno in izkazalo se je, da je postalo eno izmed najbolj uporabljenih ogrodji za izdelavo odzivnih spletnih aplikacij. Deležno je zelo velike podpore skupnosti, preko katere

lahko najdemo vse od rešitev različnih problemov, do že izdelanih tem, ki jih lahko takoj integriramo in pohitrimo celoten proces oblikovanja aplikacije. Ogradje je preprosto za uporabo in integracijo v aplikacijo ter omogoča zelo veliko možnosti prilaganja po lastnih željah. Vsebuje veliko uporabnih elementov, kot so modalna okna, opozorila, navigacijski meniji, vnosni obrazci, tabele itd. Ogradje dobiva redne nadgradnje s katerimi pridobiva vedno več dodatnih funkcionalnosti [4, 16]. Odločili smo se, da izdelamo odzivno spletno aplikacijo z uporabo tega orodja.



## **Poglavje 4      Spletno luščenje podatkov**

### **4.1    Uvod**

Internet je danes velika zbirka informacij, katerih večji del je dostopen preko svetovnega spleta. Za ogled takšnih informacij je potrebna uporaba brskalnika, ki zadostuje človeški uporabi. Ker pa je na spletu na voljo velik nabor informacij, bi bilo zelo koristno, da bi lahko stroji samodejno pridobivali takšne podatke, ki bi jih nato lahko na različne načine uporabljali za nadaljnjo uporabo.

Nekaj večjih spletnih strani danes ponuja dostop do svojih vsebin s pomočjo programskih vmesnikov, bodisi z uporabo REST- ali SOAP-protokolov. Uporabniki lahko takšne vsebine naložijo s spletnega mesta in jih uporabljajo pod določenimi pogoji, ki jih je potrebno spoštovati. Ostalim spletnim stranem oz. portalom pa večinoma ni v interesu, da bi te dragocene podatke delili, saj bi potem lahko do teh podatkov dostopali tudi drugi konkurenčni portali in na različne načine analizirali takšne podatke. To je tudi verjetno razlog, da nimajo dostopa do teh podatkov preko programskih vmesnikov.

Edina alternativna možnost za dostop do teh podatkov je preko spletnega luščenja izvorne kode HTML-strani. Je pa takšen način večkrat neustrezen, saj se struktura HTML-strani pogosto spreminja, zato lahko pridobimo napačne oz. neustrezne podatke. Spletno luščenje je metoda, ki vključuje pridobivanje strukturiranih podatkov iz spletnih strani. Spletne strani so večinoma zgrajene v označevalnih jezikih, kot sta HTML ali XHTML. Omeniti je potrebno, da so spletne strani v resnici zelo zapletene in združujejo veliko različnih HTML-elementov, ki lahko vsebujejo nepravilno gnezdene elemente ali manjkajoče oznake. Napredni brskalniki takšne napake samodejno odpravijo in prikazujejo na pravilen način. Pri spletnem luščenju pa je potrebno takšne dejavnike upoštevati. Na srečo obstaja več metod in knjižnic, ki nam to olajšajo [17]. Glede na to, da smo se odločili za izdelavo spletne aplikacije, smo se tudi osredotočili in v nadaljevanju opisali le metode in knjižnice, ki so namenjene spletnemu luščenju v programskem jeziku PHP. Na voljo je še več ostalih knjižnic, ki pa delujejo na podoben princip kot omenjene.

## 4.2 Metode in knjižnice

### 4.2.1 cURL

V programskem jeziku PHP najbolj običajna metoda za pridobivanje spletnih virov npr. spletnih strani je z uporabo cURL-knjižnice. Ko obiščemo spletno stran preko klienta, kot je npr. brskalnik, se pošlje strežniku http-zahteva, strežnik pa se nato odzove z zagotavljanjem zahtevanih virov in npr. vrne HTML-datoteko. Brskalnik nato datoteko interpretira in prikaže v uporabniku prijazni obliki. Ko naredimo cURL-zahtevo, se strežnik odziva na enak način in vrne izvorno HTML-kodo, preko katere nato lahko luščimo podatke [7]. Spodaj je podan osnovni primer cURL-zahteve (programska koda 4.1). Izvorna kodo, ki je vrnjena v rezultatu, lahko naprej obdelujemo z regularnimi izrazi (poglavje 4.2.2) ali xPath (poglavje 4.2.3), da pridobimo želene podatke.

```
// inicializacija cURL seje
$ch = curl_init('http://domena.com');

// nastavitve možnosti cURL
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// izvajanje cURL seje
$result = curl_exec($ch);

// zapiranje cURL seje
curl_close($result);

// prikaz rezultatov
print_r($result);
```

Programska koda 4.1: Primer osnovne cURL-zahteve.

### 4.2.2 Regularni izrazi

Regularni izrazi so zelo močno orodje, ki je na voljo v večini današnjih programskih jezikov. Na njih lahko gledamo kot na sistem ujemanja besedil z vzorcem. Podamo mu vzorec in besedilo ter nato uporabimo eno izmed funkcij, ki nam vrne rezultat ujemanja. V urejevalnikih besedil so nepogrešljivo orodje za iskanje, za spletno luščenje pa niso najbolj primerni, saj so veliko bolj zapleteni in težji za uporabo kot knjižnice. Dodatno nam to še otežujejo spletne strani, ki poskušajo imeti spreminjajoče se HTML-elemente. Zagotavljajo pa večjo prožnost in moč, ko se uporabljajo pravilno [17]. V primeru nepravilne uporabe pa je velika možnost, da iskani podatki ne bodo pravilni in nam bo to povzročalo le težave. Za



začetnike je takšna metoda nepriporočljiva, saj bo v večini primerov privedla do težav in izgube časa. Primer (programska koda 4.2) prikazuje osnovno uporabo regularnega izraza.

```
// besedilo v katerem iščemo vzorec
$string = "Na cesti stoji zelena žaba.";

// rezultat iskanja izbranega vzorca v besedilu
$result = preg_match('/zelena/', $string) ;
```

Programska koda 4.2: Primer uporabe regularnega izraza.

### 4.2.3 XPath in DOMXPath

XPath-poizvedovalni jezik se uporablja za navigacijo skozi elemente in attribute dokumenta XML. DOMXPath je knjižnica, ki jo vsebuje programski jezik PHP za uporabo XPath izrazov. Preneseni spletni vir spremeni v XML DOM-objekt, preko katerega s pomočjo XPath izrazov luščimo zahtevane elemente na podlagi njihovih HTML-oznak in atributov, kot so CSS-razredi. Rezultat, ki ga vrne, je seznam vozlišč ali logična vrednost določenega elementa [6]. Primer uporabe knjižnice je prikazan v sledečem primeru (programska koda 4.3).

```
// inicializacija novega DOM Document objekta
$doc = new DOMDocument;

// naložitev HTML izvirne kode pridobljene preko cURL zahteve (za primer
// smo vnesli že pridobljeno)
$doc->loadHTML('<html><body><ul id="seznam"><li>Prva vrednost</li><li>Druga
vrednost</li></ul></body></html>');

// inicializacija novega XPath DOM objekta
$xmlpath = new DOMXPath($doc);

// seznam vseh »li« vozlišč, ki so otroki vozlišča »ul«
$list = $xmlpath->query('//body//ul/li');
```

Programska koda 4.3: Primer uporabe knjižnice DOMXPath.

### 4.2.4 PHP Simple HTML DOM Parser

Je neodvisna alternativna zunanja knjižnica programskega jezika PHP, ki razvijalcem omogoča, da lahko preko PHP iščejo DOM-elemente. Iskanje elementov poteka preko selektorjev na podoben način kot v jQuery. Za njeno delovanje je potreben vsaj PHP različice 5; v konfiguraciji mora biti omogočena nastavitve »allow\_url\_fopen«, ki je potrebna za oddaljen dostop do dokumentov. Takšen način zelo poenostavi spletno luščenje podatkov [6]. V našem diplomskem delu smo se odločili za uporabo te knjižnice, saj nam je na precej hiter

način ponudila pridobivanje zelenih vrednosti. Na voljo je tudi precej dobra dokumentacija za uporabo, kar nam je še dodatno pomagalo pri odločitvi izbire te knjižnice. Nekaj osnovnih metod je prikazanih v spodnji programski kodi (programska koda 4.4).

```
// vključitev knjižnice Simple HTML DOM
include('simple_html_dom.php');

// DOM objekt iz URL naslova
$html = file_get_html('http://domena.com');

// prvi element pojavitve izbranega »id« atributa z imenom oseba
$singleFirst = $html->find('#oseba', 0);

// zadnji element pojavitve izbranega »id« atributa z imenom oseba
$singleLast = $html->find('#oseba', -1);

// vse pojavitve izbranega razreda z imenom osebe
$collection = $html->find('.osebe');
```

Programska koda 4.4: Primer uporabe knjižnice Simple HTML DOM Parser.

### 4.3 Legalnost luščanja

Danes je na voljo vedno več spletnih portalov, ki ponujajo podatke iz različnih virov, iz katerih se legalno oz. nelegalno pobirajo podatki. Vprašamo se lahko, kakšno je trenutno stanje glede takšnega početja. To je zelo zapleteno vprašanje, predvsem zaradi avtorskih pravic in intelektualne lastnine. Zakoni, ki opredeljujejo to področje, se od države do države zelo razlikujejo in zato ne obstaja jasnega odgovora. Obstaja pa nekaj univerzalnih točk, ki jih lahko najdemo za pregled takšnega početja. Večje in znane spletne strani pogosto vsebujejo dokumente, znane kot pogoji uporabe. Takšne dokumente je treba prebrati in razumeti njihove pogoje pred avtomatiziranim luščanjem vsebine. Splošni primer kršenja avtorskih pravic z luščanjem podatkov je, kadar pridobivamo podatke, ki so intelektualna lastnina nekoga drugega brez njegovega privoljenja [17].

## **Poglavje 5     Razvoj sistema**

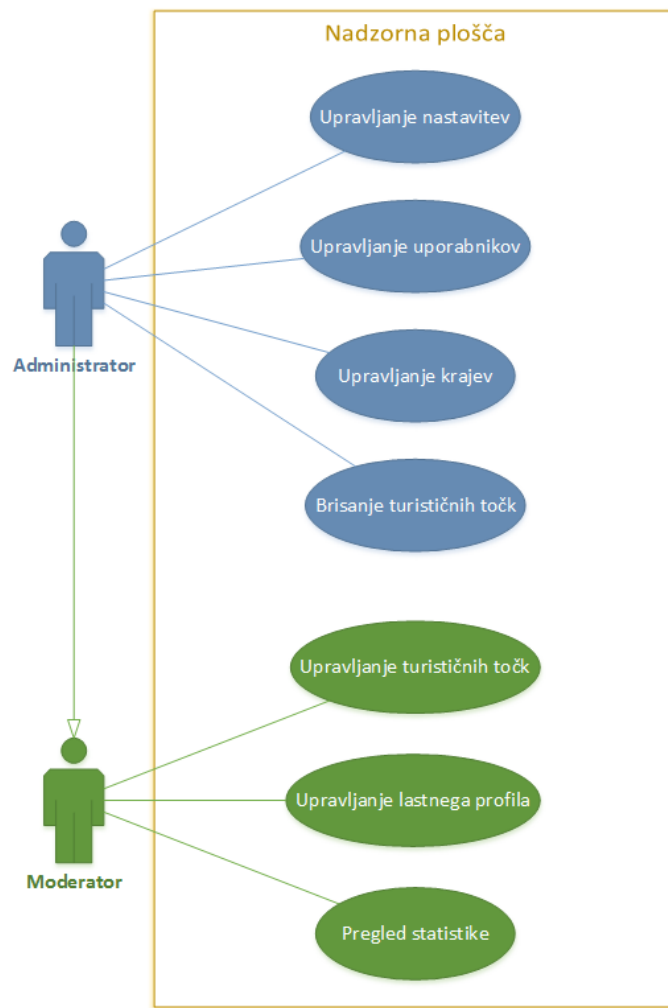
### **5.1    Načrt sistema**

#### **5.1.1 Uvod**

V tem poglavju je podrobneje opisan celoten razvoj sistema. Najprej je predstavljena arhitektura sistema ter postopek računanja globalne ocene. Sledi opis pridobivanja podatkov spletnih virov z vključenimi primeri. Nato je predstavljena s slikami in opisom celotna spletna aplikacija, ki vključuje nadzorno ploščo za upravljanje podatkov in osrednjo aplikacijo, ki služi prikazu ocen turističnih točk in krajev. Na koncu pa sledi še opis samodejnega posodabljanja ocen z uporabo systemskega klica Cron.

#### **5.1.2 Primeri uporabe**

Diagram primerov uporabe nam služi za grafično predstavitev akterjev in operacij, ki jih povezujemo z njimi. Prikazati želimo, kdo uporablja sistem in kakšni so njihovi tipični scenariji uporabe. Na takšen način si lahko lažje predstavljamo delovanje sistema in njegove funkcionalnosti. Za lažjo predstavitev smo ločili diagram na dva dela. Prvi del predstavlja nadzorno ploščo sistema (slika 5.1). Vsebuje dva akterja sistema, ki sta administrator in moderator. Administrator ima vse pravice, kot jih ima moderator, in še nekaj dodatnih, ki so dodeljene le njemu. Moderator ima možnost urejati svoj osebni profil, dodajati in urejati turistične točke, si ogledati statistiko turističnih točk in krajev. Administrator ima na voljo še dodatne pravice, kot so brisanje turističnih točk, dodajanje in urejanje krajev, dodajanje, urejanje in brisanje uporabnikov ter urejanje globalnih nastavitev spletne strani. Oba akterja pa morata biti prijavljena v sistem, da lahko uporabljata svoje funkcionalnosti.



Slika 5.1: Primeri uporabe nadzorne plošče.

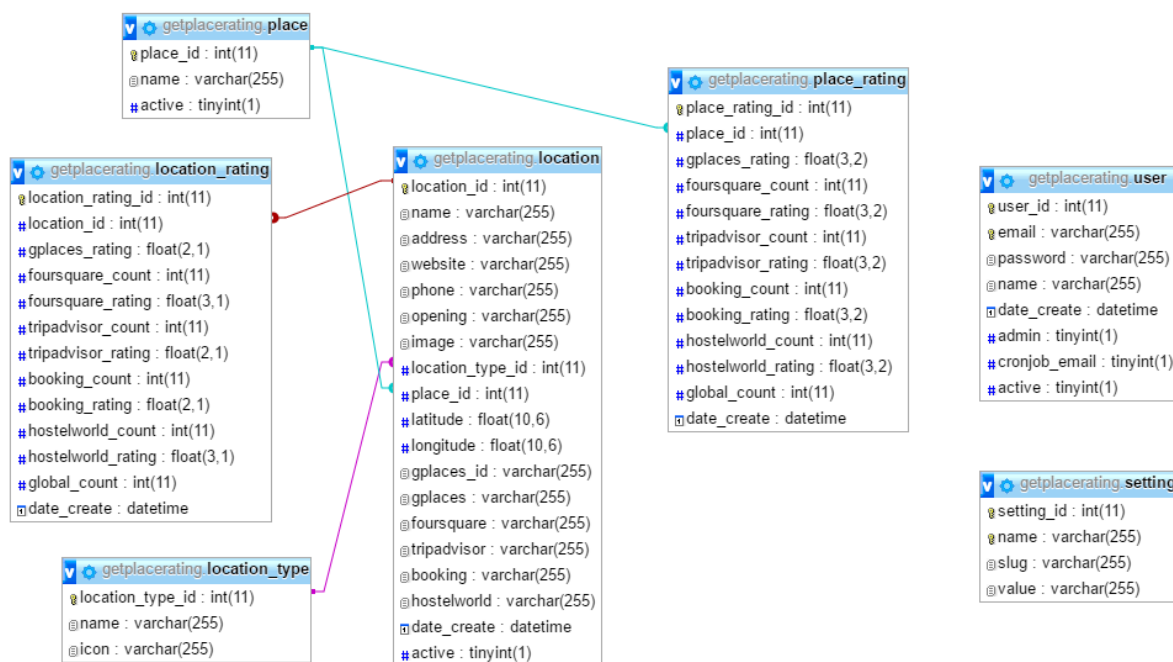
Drugi del predstavlja osrednja aplikacija, ki prikazuje akterja – obiskovalca spletne aplikacije (slika 5.2). Na voljo ima iskanje turističnih točk, pregled podrobnejših podatkov o turističnih točkah in iskanje turističnih točk preko zemljevida. Poleg tega ima na voljo prav tako iskanje krajev in pregled njihovih podrobnejših podatkov.



Slika 5.2: Primeri uporabe osrednje aplikacije.

### 5.1.3 Podatkovni model

Pri načrtovanju aplikacije je ena izmed pomembnih faz priprava pravilne strukture podatkovnega modela. Naš podatkovni model (Slika 5.3) vsebuje 4 glavne tabele (turistične točke, kraji, ocene turističnih točk, ocene krajev) in eno dodatno tabelo (kategorije turističnih točk), ki so med seboj povezane. Poleg tega imamo še 2 ločeni dodatni tabeli (uporabniki, nastavitve), ki nista neposredno povezani s prejšnjimi. Podatkovni model bi bil lahko zgrajen brez uporabe fiksnih stolpcev za vsak posamezni vir. Ker pa vsak spletni vir vsebuje posebej prilagojeno programsko kodo za spletno luščenje in bi vsak novi spletni vir zahteval temeljito predelavo celotne kode, smo se odločili za uporabo fiksnih stolpcev. Glede na to, da je rešitev na ravni prototipa, se je izkazala takšna zgradba podatkovnega modela kot dovolj dobra. V nadaljevanju pa bi kazalo razmisliti, da bi lahko naredili bolj splošen podatkovni model in s tem omogočili lažje dodajanje morebitnih novih virov.



Slika 5.3: Podatkovni model aplikacije.

Entiteta »location« hrani glavne podatke o turističnih točkah. To so ime, naslov, spletna stran, telefon, odpiralni čas, URL-povezava slike, zemljepisna širina in dolžina, URL-povezave spletnih virov, iz katerih pridobivamo podatke, datum kreiranja in ali je turistična točka aktivna in prikazana v iskalniku turističnih točk. Poleg tega imamo še dva tuja ključa, preko katerih je tabela povezana z dvema ostalima tabelama »location\_type« in »place«. Vsak zapis v to tabelo se ustvari ob kreiranju nove turistične točke. Nato pa se vnosi v tabeli za izbrano turistično točko lahko še spreminjajo ali brišejo.

Entiteta »place« hrani podatke o krajih. Vsebuje ime in ali je kraj aktiven in prikazan v iskalniku krajev. Vnos se prav tako ustvari pri kreiranju novega kraja in ga lahko naknadno spreminjamo. Brisanje zapisa pa ni mogoče, saj je vezan na tabelo »location«, ki lahko vsebuje več zapisov, vezanih na izbrani kraj.

Entiteta »location\_type« je dodatna tabela, ki hrani podatke o kategorijah turističnih točk. Vsebuje ime kategorije ter kratico (ang. slug) izbrane ikone kategorije za uporabo v programski kodi in prikaz v uporabniškem vmesniku. Te tabele ni mogoče urejati ali brisati, saj ima že določene vrednosti ter je tudi vezana na tabelo »location«, ki prav tako lahko vsebuje več zapisov, vezanih na izbrano kategorijo.

Entiteta »location\_rating« hrani podatke o ocenah turističnih točk, ki se vsakodnevno spreminjajo. Tabela vsebuje datum pridobivanja ocen, število ocen in povprečne ocene iz

posameznih spletnih virov. Poleg tega vsebuje še tuji ključ, ki je vezan na tabelo »location«. Vsak dan se za vsako turistično točko zapiše nov vnos ocen, ki je pridobljen iz drugih spletnih portalov. Vsi starejši vnosi pa nam služijo za prikaz ocen skozi čas. Vsi vnosi v tabeli so trajni in se ne brišejo ali urejajo, saj so njihovi podatki pomembni za računanje mesečnih ocen.

Entiteta »place\_rating« je skoraj identične strukture kot je vrhnja tabela le, da hrani podatke o ocenah krajev, ki se prav tako vsakodnevno spreminjajo oz. dodajajo novi zapisi za vsak kraj posebej. Vrednosti so seštevek števila ocen in njihovih povprečnih ocen pridobljenih iz podatkov vrhnje tabele »locationg\_rating«. Edini atribut, ki je tokrat drugačen, je tuji ključ, ki je vezan na tabelo »location«. Prav tako so tudi tej vnosi trajni in se ne brišejo ali urejajo, ker so prav tako potrebni za mesečno računanje ocen.

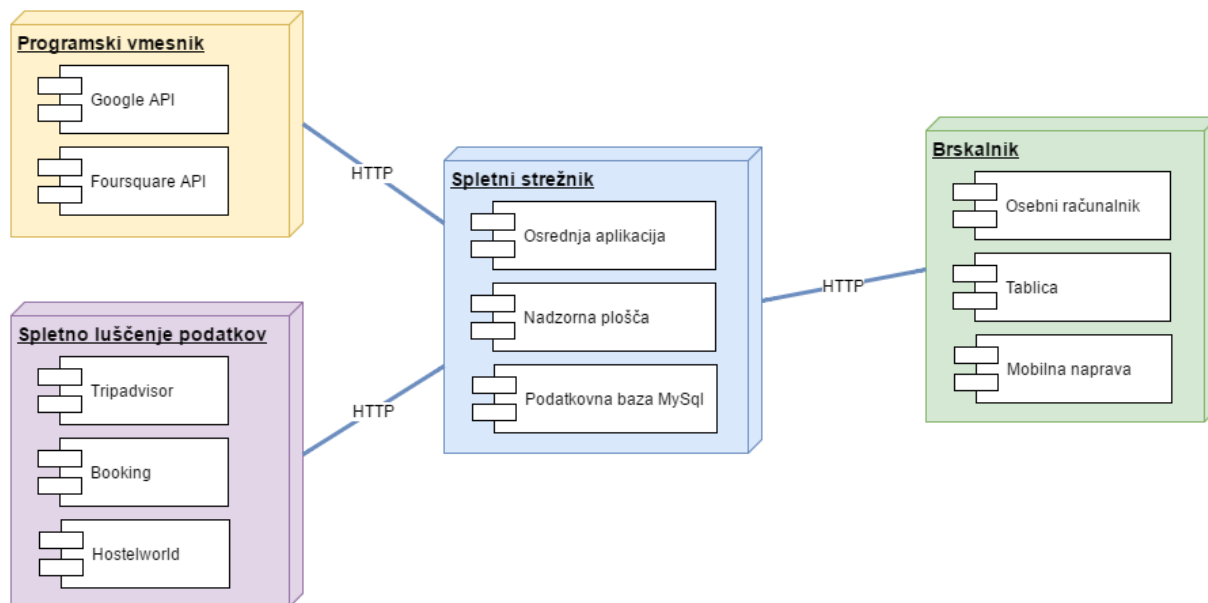
Entiteta »user« je samostojna tabela, ki hrani podatke o uporabnikih sistema. Vsebuje e-pošto in zgoščevalno geslo, ki služita za prijavo v sistem. Nato vsebuje tri vrednosti, ki določajo, ali je uporabnik aktiven oz. se lahko prijavi v sistem, ali ima status administratorja oz. moderatorja ter ali se mu pošiljajo e-obvestila o samodejnih dnevni posodobitvah ocen turističnih točk in krajev. Poleg tega pa tabela vsebuje še ime uporabnika in datum kreiranja. Vnos v tabelo se izvede ob kreiranju novega uporabnika. Naknadno se tej vnosi lahko urejajo in brišejo.

Entiteta »setting« je prav tako samostojna tabela, ki služi hranjenju globalnih vrednosti sistema. Vsebuje ime, vrednost in kratico (ang. slug) preko katere pridobivamo v vrednosti v programski kodi. Tabela ima že vnaprej določene vrednosti, zato je ni mogoče spreminjati.

Vse entitete vsebujejo primarne ključe, ki se z novimi vnosi samodejno povečujejo. Poleg tega nekatere entitete vsebujejo tudi dodatne indekse na določenih atributih, ki so namenjeni hitrejšemu procesiranju poizvedb v tabelah podatkovne baze.

#### **5.1.4 Implementacijski pogled**

Implementacijski pogled je namenjen prikazu komponent, ki jih vsebuje sistem. Na sliki (slika 5.4) so prikazane s pomočjo komponentnega diagrama. V našem sistemu lahko z brskalnikom preko različnih naprav dostopamo do spletnega strežnika, ki nam prikazuje osrednjo aplikacijo in nadzorno ploščo, ki vsebuje podatke iz podatkovne baze. Spletni strežnik dodatno komunicira s programskimi vmesniki. Povezan pa je tudi na izbrane spletne strani, namenjene luščenju podatkov.



Slika 5.4: Implementacijski pogled sistema.

### 5.1.5 Postopek računanja globalne ocene

Globalne ocene turističnih točk in krajev so izračunane in obarvane glede na trenutni trend, ki temelji na rezultatu izračuna njunih formul. Za izračun so uporabljene vrednosti zadnjega dneva v mesecu. Ko je rezultat večji ali enak kot prejšnji mesec, se ocena obarva v zeleni barvi. V primeru manjšega rezultata pa je ocena obarvana v rdečo barvo. Prvi mesec je ocena vedno obarvana v zeleno barvo, nato pa se spreminja glede na trend. Mesečni trend globalnih ocen je prikazan na grafu.

Formula globalne ocene turistične točke (5.1):

- $P_t$  – Povprečna ocena izbrane turistične točke na posameznem spletnem viru.
- $\check{S}_t$  – Št. ocen izbrane turistične točke na posameznem spletnem viru.
- $G_t$  – Globalni seštevek ocen izbrane turistične točke iz vseh spletnih virov.
- $O_t$  – Globalna ocena turistične točke.

$$O_t = \sum_{i=1}^n P_{t_i} \frac{\check{S}_{t_i}}{G_{t_i}} = P_{t_1} \times \frac{\check{S}_{t_1}}{G_{t_1}} + \dots + P_{t_n} \times \frac{\check{S}_{t_n}}{G_{t_n}} \quad (5.1)$$



Formula globalne ocene kraja (5.2):

- $P_k$  – Povprečna ocena vseh turističnih točk izbranega kraja na posameznem spletnem viru.
- $\check{S}_k$  – Št. ocen vseh turističnih točk izbranega kraja na posameznem spletnem viru.
- $G_k$  – Globalni seštevek števila ocen vseh turističnih točk kraja iz vseh spletnih virov.
- $O_k$  – Globalna ocena kraja.

$$O_k = \sum_{i=1}^n P_{k_i} \frac{\check{S}_{k_i}}{G_{k_i}} = P_{k_1} \times \frac{\check{S}_{k_1}}{G_{k_1}} + \dots + P_{k_n} \times \frac{\check{S}_{k_n}}{G_{k_n}} \quad (5.2)$$

## 5.2 Pridobivanje podatkov

### 5.2.1 Google API

Glede na to, da Google ponuja veliko storitev, ki so na voljo brezplačno, se nam je zdela smiselna uporaba le-teh. Preko storitve Google Maps omogoča na zemljevidu pregledovanje podatkov turističnih točk in njihovih ocen, ki so jih podali uporabniki. Ker vsebuje ogromno bazo, nam je to zelo pomagalo pri pridobivanju podatkov turističnih točk. Za naše potrebe smo izbrali programski vmesnik Google Places API Web Service, ki omogoča pridobivanje podatkov o turističnih točkah. Za pridobitev podrobnosti o turistični točki je treba narediti zahtevo na API, ki nato vrne podatke v JSON-formatu (programska koda 5.1).

```
{
  "html_attributions" : [],
  "result" : {
    "formatted_address" : "48 Pirrama Road, Pyrmont NSW, Australia",
    "formatted_phone_number" : "(02) 9374 4000",
    "geometry" : {
      "location" : {
        "lat" : -33.8669710,
        "lng" : 151.1958750
      },
      "viewport" : {
        "northeast" : {
          "lat" : -33.8665053,
          "lng" : 151.1960371
        }
      }
    }
  }
}
```

```

        "southwest" : {
            "lat" : -33.8669293,
            "lng" : 151.1952183
        }
    },
    "icon" :
"http://maps.gstatic.com/mapfiles/place_api/icons/generic_business-71.png",
    "id" : "4f89212bf76dde31f092cfc14d7506555d85b5c7",
    "international_phone_number" : "+61 2 9374 4000",
    "name" : "Google Sydney",
    "place_id" : "ChIJN1t_tDeuEmsRUsoyG83frY4",
    "scope" : "GOOGLE",
    "alt_ids" : [
        {
            "place_id" : "D9iJyWEHuEmuEmsRm9hTkapTCrk",
            "scope" : "APP"
        }
    ],
    "rating" : 4.70
},
"status" : "OK"
}

```

Programska koda 5.1: Izsek JSON-odgovora na podano Google API-zahtevo.

Za pridobivanje podatkov o turistični točki (programska koda 5.2) je bilo potrebno poslati parameter »placeid«, ki ga pridobimo iz podatkovne baze. Podatki turistične točke, ki smo jih želeli pridobiti, so povprečna ocena, naslov, telefon, odpiralni čas, Google Maps povezava, domača spletna stran in slika. Želeli smo uporabiti tudi povprečne ocene turističnih točk, a smo se soočili s težavo. Odgovor na zahtevo, ki ga vrne programski vmesnik, ne vsebuje števila ocen. Torej bi lahko ta podatek v primerih, ko je povprečna ocena sestavljena iz premalo ocen, izstopal iz povprečja in bi lahko zelo vplival na globalno oceno turističnih točk in krajev. Zato smo se odločili, da uporabimo le ostale podatke, ki so na voljo.

```

$json =
file_get_contents('https://maps.googleapis.com/maps/api/place/details/json?
placeid=' . $_place_id . '&language=sl&key=' . $this->google_key);

$objj = json_decode($json, true);

$result = array(
    'countReviews'    => 0,
    'ratingReviews'   => isset($obj['result']['rating']) ?
    $obj['result']['rating'] : 0,

```

```

    'address'          => isset($obj['result']['formatted_address']) ?
$obj['result']['formatted_address'] : "",
    'phone'            => isset($obj['result']['formatted_phone_number'])
? $obj['result']['formatted_phone_number'] : "",
    'opening'          =>
isset($obj['result']['opening_hours']['weekday_text']) ?
$obj['result']['opening_hours']['weekday_text'] : "",
    'gmaps'            => isset($obj['result']['url']) ?
$obj['result']['url'] : "",
    'website'          => isset($obj['result']['website']) ?
$obj['result']['website'] : "",
    'image_reference'  =>
isset($obj['result']['photos'][0]['photo_reference']) ?
$obj['result']['photos'][0]['photo_reference'] : ""
);

$imageUrl =
'https://maps.googleapis.com/maps/api/place/photo?maxwidth=400&maxheight=250&photoreference=' . $result['image_reference'] . '&key=' . $this->google_key;

$result['image'] = $imageUrl;

```

Programska koda 5.2: Pridobivanje podatkov z Google API.

### 5.2.2 Foursquare API

Foursquare je eno izmed najbolj priljubljenih geolokacijskih družabnih omrežij. Vsebuje tudi večjo bazo podatkov o turističnih točkah. Uporabnikom je omogočeno urejanje podatkov turističnih točk, ocenjevanje in pisanje mnenj. Ocenjevanje poteka po posebnem algoritmu, ki na podlagi razmerja podanih mnenj in všečkov izračuna oceno. Uporablja ga veliko uporabnikov, zato smo se odločili, da ga vključimo na naš seznam. Poleg ocen imamo na voljo tudi več podrobnejših podatkov o turističnih točkah, ki smo jih lahko združili s podatki, ki jih ponuja Google. Za dostop do teh podatkov nam je na voljo programski vmesnik Foursquare API, ki deluje na podoben način kot Google API. Za dostop do podatkov turistične točke je prav tako treba narediti API-zahtevo, ki nato vrne odgovor na zahtevo (programska koda 5.3), ki ga pretvorimo v JSON-format za nadaljnje obdelovanje.

```

response: {
  venue: {
    id: "4bb11095f964a52042773ce3"
    name: "Prešernov trg"
    contact: { }
    location: {
      address: "Prešernov trg 1"
    }
  }
}

```

```

lat: 46.051483462176165
lng: 14.506030082702637
labeledLatLngs: [
  {
    label: "display"
    lat: 46.051483462176165
    lng: 14.506030082702637
  }
]
postalCode: "1000"
cc: "SI"
city: "Ljubljana"
state: "Ljubljana"
country: "Slovenia"
formattedAddress: [
  "Prešernov trg 1"
  "1000 Ljubljana"
]
}
canonicalUrl: "https://foursquare.com/v/pre%C5%A1ernov-
trg/4bb11095f964a52042773ce3"
url: "http://www.visitljubljana.com"
rating: 9.2
ratingColor: "00B551"
ratingSignals: 213
}
}

```

Programska koda 5.3: Izsek odgovora na podano Foursquare API-zahtevo.

Za izvedbo zahteve (programska koda 5.4) je treba poslati parameter »venueid«, ki ga pridobimo preko URL-povezave, shranjene v tabeli podatkovne baze. Povezava ima naslednjo strukturo: »https://foursquare.com/v/pre%C5%A1ernov-trg/4bb11095f964a52042773ce3«. Ker se parameter »venueid« nahaja v zadnjem delu povezave, je bilo treba povezavo najprej razbiti na več delov in izbrati zadnji del, kjer se nahaja parameter, ter ga nato poslati v zahtevi. Podatki turistične točke, ki smo jih želeli pridobiti, so povprečna ocena, število ocen, domača spletna stran in slika. Spletno stran in sliko nato uporabimo v primeru, ko nista bili pridobljeni preko Google API.

```

$url = parse_url($url);
$path = explode("/", $url['path']);
$name = str_replace("-", " ", ucwords($path[2]));
$venueId = $path[3];

```

```

$json = file_get_contents('https://api.foursquare.com/v2/venues/' .
$venueId . '?client_id=' . $this->foursquare_client_id . '&client_secret='
. $this->foursquare_client_secret . '&v=20160804');
$obj = json_decode($json, true);

$result = array(
    'countReviews' => isset($obj['response']['venue']['ratingSignals']) ?
$obj['response']['venue']['ratingSignals'] : 0,
    'ratingReviews' => isset($obj['response']['venue']['rating']) ?
$obj['response']['venue']['rating'] : 0,
    'website'      => isset($obj['response']['venue']['url']) ?
$obj['response']['venue']['url'] : "",
    'image'        =>
isset($obj['response']['venue']['bestPhoto']['prefix']) ?
$obj['response']['venue']['bestPhoto']['prefix'] . "400x250" .
$obj['response']['venue']['bestPhoto']['suffix'] : ""
);

```

Programska koda 5.4: Pridobivanje podatkov s Foursquare API.

### 5.2.3 Tripadvisor

Tripadvisor je največji in verjetno najbolj poznan turistični portal. Ponuja veliko različnih pripomočkov, ki turistom zelo pomagajo pri načrtovanju potovanj oz. izbiri destinacij. Med njimi najdemo primerjalnik cen nočitev, iskalnik letalskih povezav, turistične vodiče in še veliko drugih možnosti. Poleg tega vsebuje ogromno število ocen in mnenj o različnih turističnih točkah. Vrednosti ocen so izračunane po posebnem algoritmu, ki upošteva več dejavnikov, kot so število ocen, priljubljenost ocen in starost ocen. V pomoč nam je bil za pridobivanje ocen za vse naše kategorije turističnih točk.

Spletni portal ponuja možnost uporabe podatkov preko njihovega programskega vmesnika. Za dostop je treba zaprositi in pogoji za dodelitev dostopa so zelo strogi in zahtevni. Zato nam žal kljub večkratnemu poskušanju ni uspelo pridobiti dostopa. Ker pa nam je ta spletni portal zelo ustrezal zaradi ogromne baze števila ocen, smo se odločili, da bomo podatke pridobivali preko spletnega luščenja strani (programska koda 5.5).

```

<div class="rs rating" property="aggregateRating" typeof="AggregateRating">
    <span class="rate sprite-rating_rr rating_rr"> 
    </span>

```

```
<a class="more taLnk"
onclick="ta.trackEventOnPage('Ratings_Info','review_count','');"
href="#REVIEWS" property="reviewCount" content="707">707 Reviews</a>
</div>
```

Programska koda 5.5: Izsek izvirne kode spletnega portala Tripadvisor.

V podatkovni bazi imamo v tabeli shranjeno URL-povezavo, preko katere nato z luščenjem dobimo vrednosti. Povezava ima naslednjo strukturo: »https://www.tripadvisor.com/Hotel\_Review-g274873-d276339-Reviews-Hotel\_Lev\_Ljubljana-Ljubljana\_Upper\_Carniola\_Region.html«. Vrednosti ocen se nahajajo na začetni strani, zato ni bilo potrebe po dodatnem spreminjanju URL-povezave. Preveriti pa je bilo potrebno, ali je povezava pravilna. Nato smo pregledali, kakšna je HTML-struktura spletnega portala in kje se nahajata vrednosti o številu ocen in o povprečni oceni. Ugotovili smo, da se povprečna ocena nahaja v atributu od slike, število ocen pa v atributu povezave. Oba atributa sta unikatna, zato nam ni predstavljalo težav pri luščenju vrednosti (programska koda 5.6). Povprečno oceno smo pridobili preko iskanja atributa »property« v HTML-elementu »a«. Nato pa znotraj elementa pridobili še vsebino, ki se nahaja v elementu »property«, kar je v našem primeru povprečna ocena. Na podoben način smo pridobili tudi število ocen, le da smo tokrat namesto iskanja v elementu »a« iskali vsebino v elementu »img«.

```
$html = file_get_html($_url);

$countReviews = $html->find('a[property=reviewCount]', 0)->content;
$ratingReviews = $html->find('img[property=ratingValue]', 0)->content;

$result = array(
    'countReviews' => $countReviews,
    'ratingReviews' => $ratingReviews
);
```

Programska koda 5.6: Pridobivanje vrednosti ocen iz spletnega portala Tripadvisor.

### 5.2.4 Booking

Ker je Booking.com ena izmed najpomembnejših turističnih spletnih strani za rezervacijo nočitev, je bila logična odločitev, da jo izberemo za pridobivanje ocen o prenočitvah ter s tem zadostimo našo kategorijo o prenočitvah. Ocene so načeloma zelo relevantne, saj deluje po principu, da lahko oceno dodajo le uporabniki, ki so dejansko izvedli rezervacijo. Poleg tega pa upravljalci strani dodatno preverijo vsako oceno in njene komentarje preden je objavljena.

Spletna stran nam žal ne ponuja programskega vmesnika za dostop do podatkov. Zaradi tega smo bili primorani podatke pridobivati preko spletnega luščenja strani (programska koda 5.7).

```
<div class="review_list_score_breakdown_left">
    <h4 class="review_list_score_title">Review score</h4>
    <p id="review_list_score_count" class="review_list_score_count">
        Based on <strong>1531 reviews</strong>
    </p>
    <div id="review_list_main_score" class="review_list_score">
        8.6
    </div>
</div>
```

Programska koda 5.7: Izsek izvirne kode spletne strani Booking.

Ponovno smo iz shranjene URL-povezave v tabeli podatkovne baze pridobili povezavo, ki je zgrajena na naslednji način: »<http://www.booking.com/hotel/si/lev.html>«. Vrednosti ocen se nahajajo na podstrani »#tab-reviews«. Zato smo najprej preverili, ali je povezava ustrezna, in ji nato priključili na konec URL-povezave še podstran. Nato smo na že znani način pregledali HTML-strukturo strani in ugotovili, kje se nahaja število ocen in povprečna ocena. Ponovno sta obe vrednosti v unikatnih HTML-atributih, kar nam omogoči lažje pridobivanje njunih vrednosti (programska koda 5.8). Število ocen smo pridobili tako, da smo najprej poiskali HTML-oznako prvega elementa »p«, ki vsebuje atribut »id« z določenim imenom. Nato smo znotraj tega elementa poiskali oznako »strong« ter pridobili golo besedilo, ki se nahaja v elementu. Ker pa smo pridobili poleg števila ocen tudi besedilo, smo to rešili s PHP-funkcijo, ki nam iz besedila vrne le številsko vrednost. Povprečno oceno pridobimo še na lažji način, saj se nahaja že v prvem elementu, zato kar poiščemo HTML-oznako prvega elementa »div«, ki ima atribut »id« z določenim imenom. Nato pa preprosto pridobimo golo besedilo tega elementa.

```
$html = file_get_html($_url . "#tab-reviews");

$countReviewsString = $html->find('p[id=review_list_score_count]', 0)-
>find('strong', 0)->plaintext;
$countReviews = filter_var($countReviewsString,
FILTER_SANITIZE_NUMBER_INT);
$ratingReviews = $html->find('div[id=review_list_main_score]', 0)-
>plaintext;

$result = array(
    'countReviews' => $countReviews,
    'ratingReviews' => $ratingReviews
);
```

Programska koda 5.8: Pridobivanje vrednosti ocen iz spletne strani Booking.

### 5.2.5 Hostelworld

Hostelworld spada med glavne spletne strani za rezervacijo nočitev v hostlih na vsem svetu. Ocene so zelo relevantne, saj lahko ocenjujejo le turisti, ki so rezervirali hostel preko te spletne strani. To sta tudi glavna razloga, da smo to spletno stran dodali na naš seznam pridobivanja ocen ter s tem še dodatno zadostili kategorijo prenočišč.

Kot že na ostalih dveh je bilo tudi tukaj potrebno pridobivanje podatkov preko spletnega luščenja. Na našo srečo nam je bila HTML-struktura strani (programska koda 5.9) ponovno prav tako naklonjena in smo lahko preprosto pridobili izbrane vrednosti ocen.

```
<div class="microsite_rating clearfix ">
  <div class="ratingbox">
    8.5
  </div>
  <div class="ratingdetails">
    <span class="ratingword">Fabulous</span>
    <span class="reviews-count">
      <i class="fa fa-comment-o"></i> 2,212 Total Reviews</span>
    </div>
  </div>
```

Programska koda 5.9: Izsek izvirne kode spletne strani Hostelworld.

Ponovno smo morali URL-povezavi za pridobivanje vrednosti (programska koda 5.10), ki jo imamo v tabeli podatkovne baze, priključiti podstran »#tab-reviews«, kjer se nahajajo vrednosti ocen. Kot nam je že znano, smo morali tudi preveriti pravilnost podane povezave, preden smo povezavi priključili podstran. Nato smo najprej pridobili število ocen tako, da najprej poiščemo prvi HTML-element »span«, ki vsebuje CSS-razred z določenim imenom. Potem pridobimo golo besedilo in ga ponovno preko PHP-funkcije prečistimo, da dobimo le številsko vrednost. Poprečno oceno poiščemo s HTML-elementom »div«, ki vsebuje CSS-razred z izbranim imenom. Ponovno pridobimo oceno preko golega besedila.

```
$html = "";
if (substr($_url, -1) == "/") {
    $html = file_get_html($_url . 'reviews');
} else {
    $html = file_get_html($_url . '/reviews');
}

$html = file_get_html($_url . "#tab-reviews");
```



```

$countReviewsString = $html->find('span[class=reviews-count]', 0)-
>plaintext;
$countReviews = filter_var($countReviewsString,
FILTER_SANITIZE_NUMBER_INT);
$ratingReviews = $html->find('div[class=ratingbox]', 0)->plaintext;

$result = array(
    'countReviews' => $countReviews,
    'ratingReviews' => $ratingReviews
);

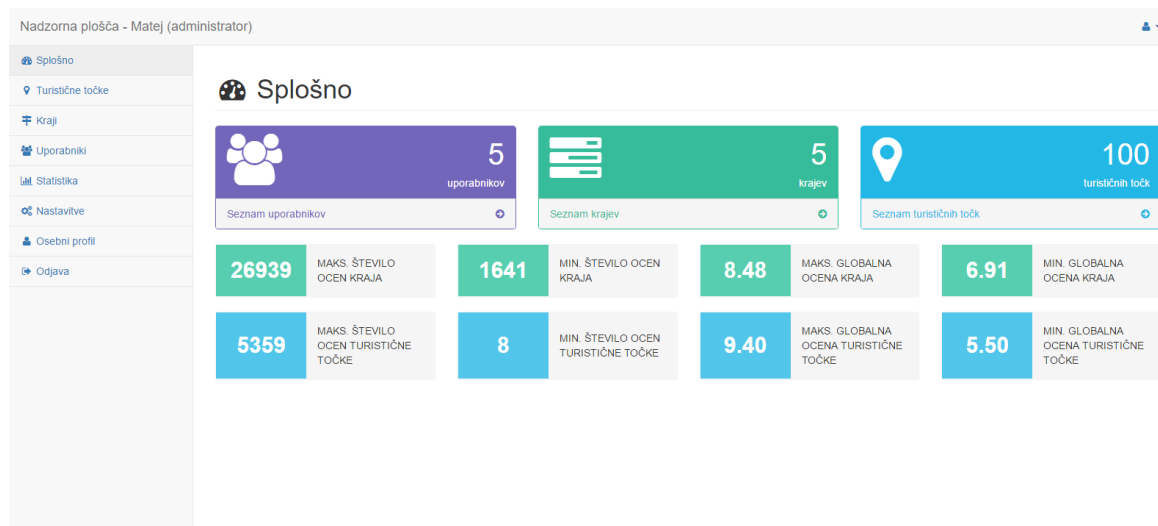
```

Programska koda 5.10: Pridobivanje vrednosti ocen iz spletne strani Hostelworld.

## 5.3 Pregled spletne aplikacije

### 5.3.1 Nadzorna plošča

Nadzorna plošča (slika 5.5) temelji na predelani grafični obliki Bootstrap predloge SB Admin 2, ki vključuje že veliko uporabnih komponent, kot so opozorila, grafi, tabele itd. Za uporabo nadzorne plošče je potrebna prijava, saj je namenjena le uporabnikom, ki so dodani v sistemu. Na levi strani je postavljen glavni meni, preko katerega dostopamo do različnih podstrani za upravljanje in ogled podatkov. Prva podstran nadzorne plošče vsebuje nekaj splošnih podatkov sistema, kot so število uporabnikov, število krajev in število turističnih točk. Poleg tega pa je prikazanih še nekaj statističnih podatkov o ocenah krajev in turističnih točk.



Slika 5.5: Izgled nadzorne plošče.

### 5.3.1.1 Pravice uporabnikov

Glede na določene pravice se uporabnikom prikazujejo določene funkcionalnosti nadzorne plošče. Na voljo sta dva različna tipa uporabnikov, to sta moderator in administrator.

Pravice moderatorjev:

- dodajanje in urejanje turističnih točk,
- ogled statistike,
- urejanje osebnega profila.

Pravice administratorjev (vključujejo vse pravice moderatorjev):

- brisanje turističnih točk,
- dodajanje in urejanje krajev,
- dodajanje, urejanje in brisanje uporabnikov,
- urejanje globalnih nastavitev.

### 5.3.1.2 Upravljanje turističnih točk, krajev in uporabnikov

Upravljanje turističnih točk, krajev in uporabnikov poteka na podoben princip. Osredotočili smo se na predstavitev upravljanja turističnih točk, ki je največkrat uporabljeno. Seznam turističnih točk je prikazan v tabeli (slika 5.6) in prikazuje nekaj glavnih atributov turističnih točk, kot so ime, tip, kraj, zemljepisna širina, zemljepisna dolžina, čas kreiranja in ali je turistična točka aktivna oz. prikazana v osrednji aplikaciji. Desno na vrhu imamo možnost dodajanja nove turistične točke. V tabeli pa preko klika na ikone v zadnjem stolpcu lahko turistično točko pregledamo, uredimo ali izbrišemo.

Nadzorna plošča - Matej (administrator)

Splošno  
 Turistične točke  
 Kraji  
 Uporabniki  
 Statistika  
 Nastavitve  
 Moji podatki  
 Odjava

## Turistične točke

+ Dodaj novo turistično točko

Seznam turističnih točk

Prikaži 10 zapisov Išči:

Ime	Tip	Kraj	Zemlj. širina	Zemlj. dolžina	Čas kreiranja	Aktivno	
Austria Trend Hotel	Prenočišča	Ljubljana	46.081711	14.513525	10.8.2016 19:46	☑	🔍 ✎ ✖
Best Western Plus Hotel Piramida	Prenočišča	Maribor	46.559242	15.653603	10.8.2016 23:07	☑	🔍 ✎ ✖
Best Western Premier Hotel Slon	Prenočišča	Ljubljana	46.052517	14.503793	10.8.2016 19:38	☑	🔍 ✎ ✖
Bit Center hotel	Prenočišča	Ljubljana	46.048656	14.549145	10.8.2016 21:41	☑	🔍 ✎ ✖
Casino Park	Atrakcije	Nova Gorica	45.954994	13.645247	12.8.2016 18:14	☑	🔍 ✎ ✖
Casino Perla	Atrakcije	Nova Gorica	45.959084	13.649249	12.8.2016 18:16	☑	🔍 ✎ ✖
Citycenter Celje	Nakupovanje	Celje	46.241550	15.277393	12.8.2016 17:08	☑	🔍 ✎ ✖
DIC hostel	Prenočišča	Ljubljana	46.048275	14.518657	10.8.2016 21:42	☑	🔍 ✎ ✖
Europark Maribor	Nakupovanje	Maribor	46.553978	15.653144	10.8.2016 22:38	☑	🔍 ✎ ✖
Fabrika	Hrana in pijača	Nova Gorica	45.957119	13.647187	12.8.2016 17:45	☑	🔍 ✎ ✖

Prikazujem 1 do 10 od 100 zapisov

Pred 1 2 3 4 5 ... 11 Nasl.

Slika 5.6: Seznam turističnih točk v tabeli.

V primeru izbire dodajanja nove turistične točke oz. klika ikone za urejanje se nam odpre podstran (slika 5.7), kjer lahko dodajamo oz. urejamo turistično točko. Poleg že prej omenjenih atributov imamo na voljo še vnosna polja za vnos URL-povezav spletnih virov, preko katerih pridobivamo podatke o turističnih točkah in ocenah. Za lažje iskanje in preverjanje pravilnega naslova turistične točke je na voljo zemljevid, ki avtomatično pridobi potrebne vrednosti atributov in jih doda v zatemnjena vnosna polja.

## Turistična točka - Austria Trend Hotel

Podatki turistične točke

Ime: Austria Trend Hotel

Tip: 

Prenočišča

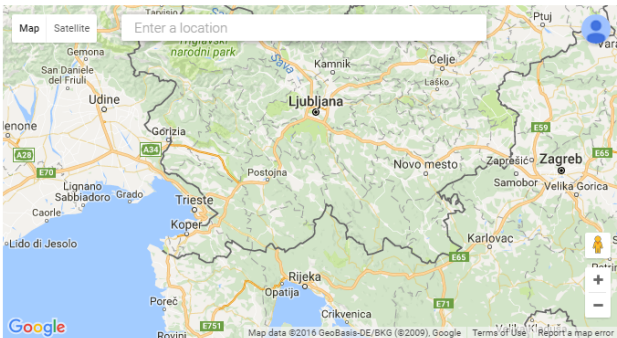
Kraj: 

Ljubljana

Gmaps:

Map Satellite

Enter a location



Google place id: ChIJ98OKAegyZUcRbPMux3lw1\_0

Zemlj. širina: 46.081711

Zemlj. dolžina: 14.513525

FourSquare: https://foursquare.com/v/austria-trend-hotel/4b89292ff964a520232032e3

TripAdvisor: https://www.tripadvisor.com/Hotel\_Review-g274873-d291456-Reviews-Austria\_Trend\_Hotel\_Ljubljana-

Booking: http://www.booking.com/hotel/si/ljubljana.html

HostelWorld: http://

Aktivno: 

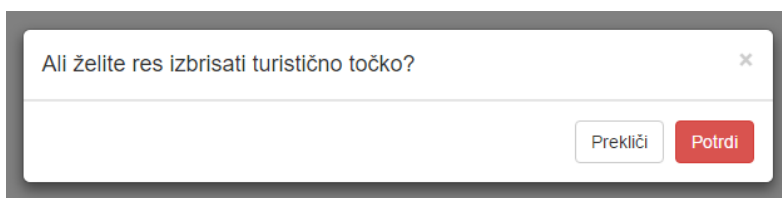
DA

Spremeni podatke

Nazaj

Slika 5.7: Urejanje izbrane turistične točke.

V tabeli se nam ob kliku ikone za izbris turistične točke prikaže opozorilno okno (slika 5.8) za potrditev oz. preklic izbrisa.



Ali želite res izbrisati turistično točko?

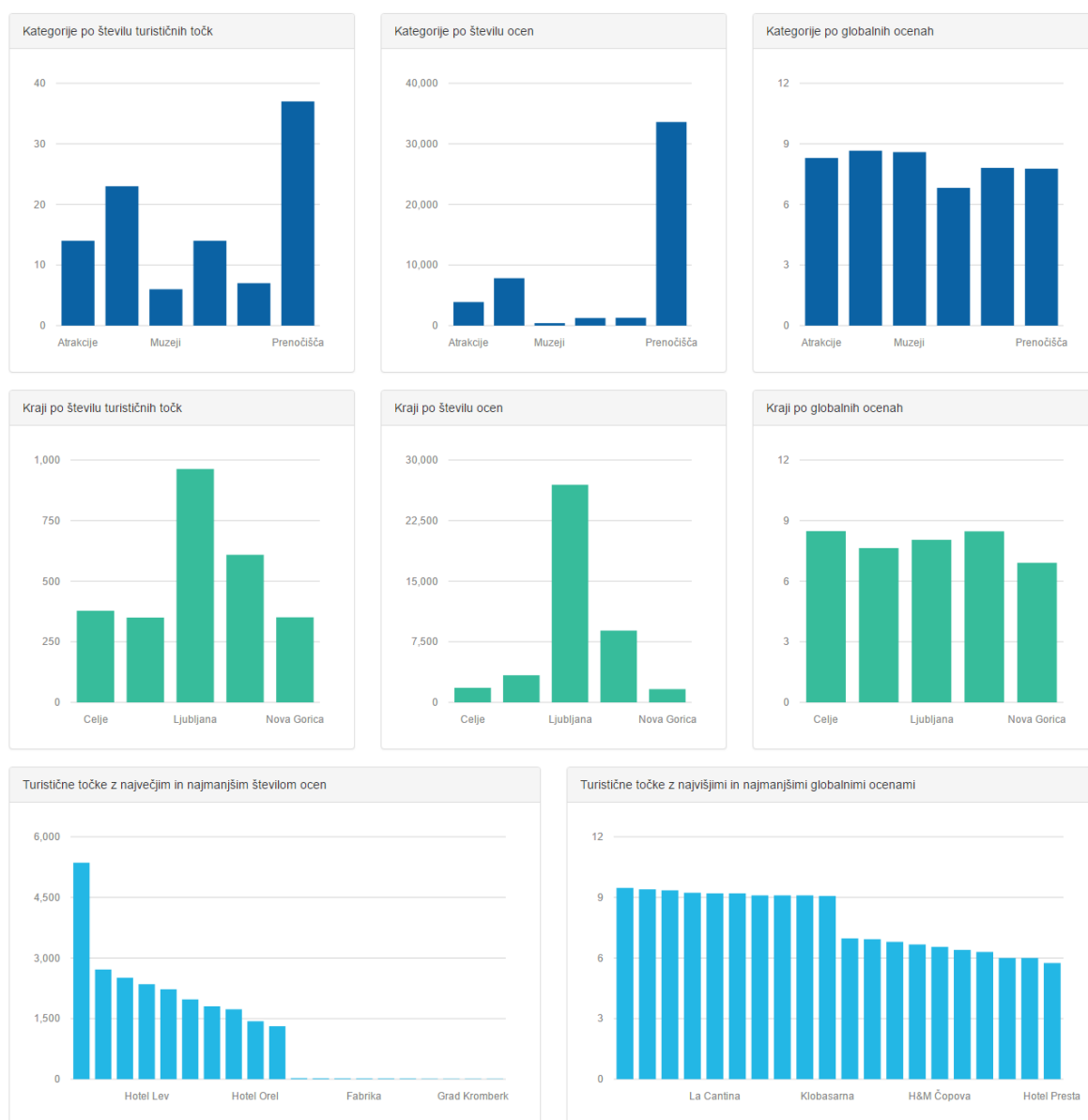
Prekliči Potrdi

Slika 5.8: Opozorilno okno izbrisa turistične točke.

### 5.3.1.3 Statistika

Poleg upravljanja podatkov je v nadzorni plošči na voljo pregled statistike podatkov v treh skupinah grafov (slika 5.9). Prva skupina grafov predstavlja grafe kategorij. Naslednja skupina prikazuje grafe krajev. Zadnja skupina pa prikazuje grafe turističnih točk.

#### Statistika

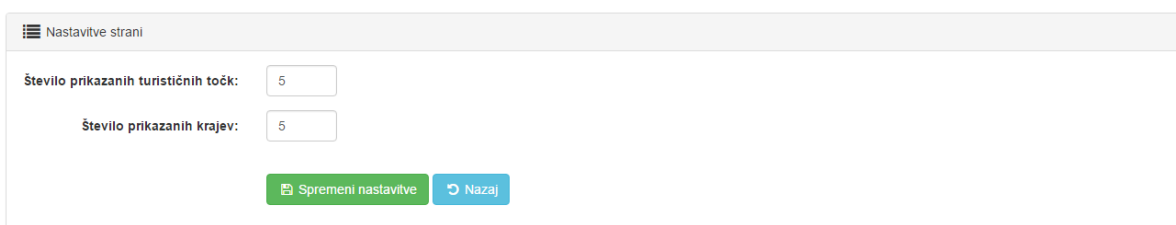


Slika 5.9: Grafi za prikazovanje statistike.

### 5.3.1.4 Nastavitve

Za upravljanje globalnih nastavitev osrednje aplikacije je na voljo posebna stran (slika 5.10) z nastavitvami. Nastavljamo lahko število prikazov elementov v seznamu iskalnika glede na paginacijo. Lahko se nastavi tako za turistične točke kot tudi za kraje.

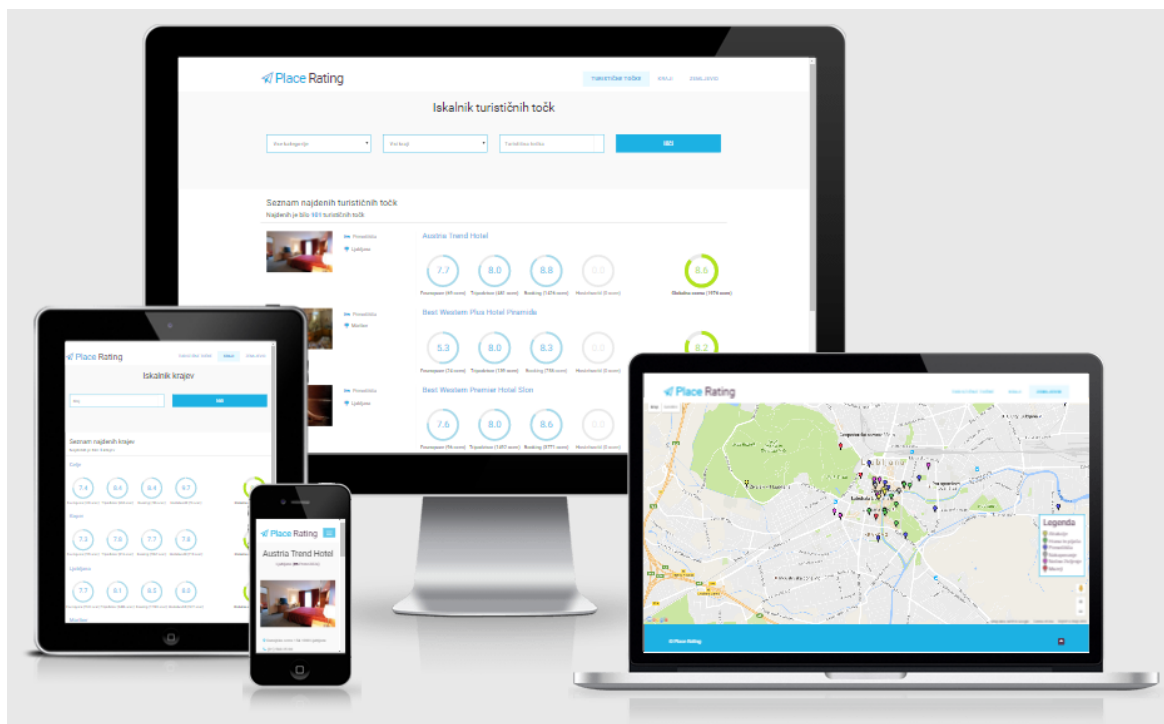
#### Nastavitve



Slika 5.10: Globalne nastavitve osrednje aplikacije.

### 5.3.2 Osrednja aplikacija

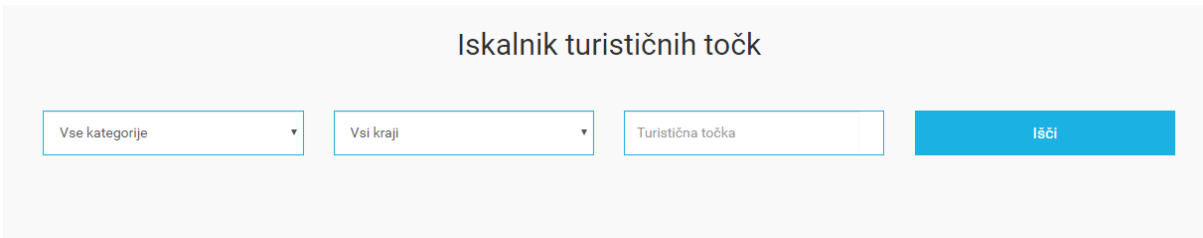
Kot že za nadzorno ploščo smo se tudi za osrednjo aplikacijo (Slika 5.11) odločili za izbiro Bootstrap ogrodja za prikaz odzivne oblike aplikacije. Zgrajena je zelo minimalistično, saj smo želeli, da je uporabniška izkušnja preprosta in praktična. Na vrhu imamo meni, ki vsebuje tri glavne podstrani, to so turistične točke, kraji ter zemljevid.



Slika 5.11: Izgled osrednje aplikacije.

### 5.3.2.1 Turistične točke

Na podstrani turistične točke imamo na voljo iskalnik (slika 5.12), ki ponuja možnost izbire iskanja glede na kategorijo, kraj in določen niz, ki lahko vsebuje del imena turistične točke. Lahko izberemo oz. napišemo iskalne parametre ali pa preprosto le kliknemo na gumb »išči«, ki izvede iskanje. V primeru, da nimamo izbranih nobenih parametrov, bo prikazan seznam vseh turističnih točk (slika 5.13), ki jih imamo na voljo.



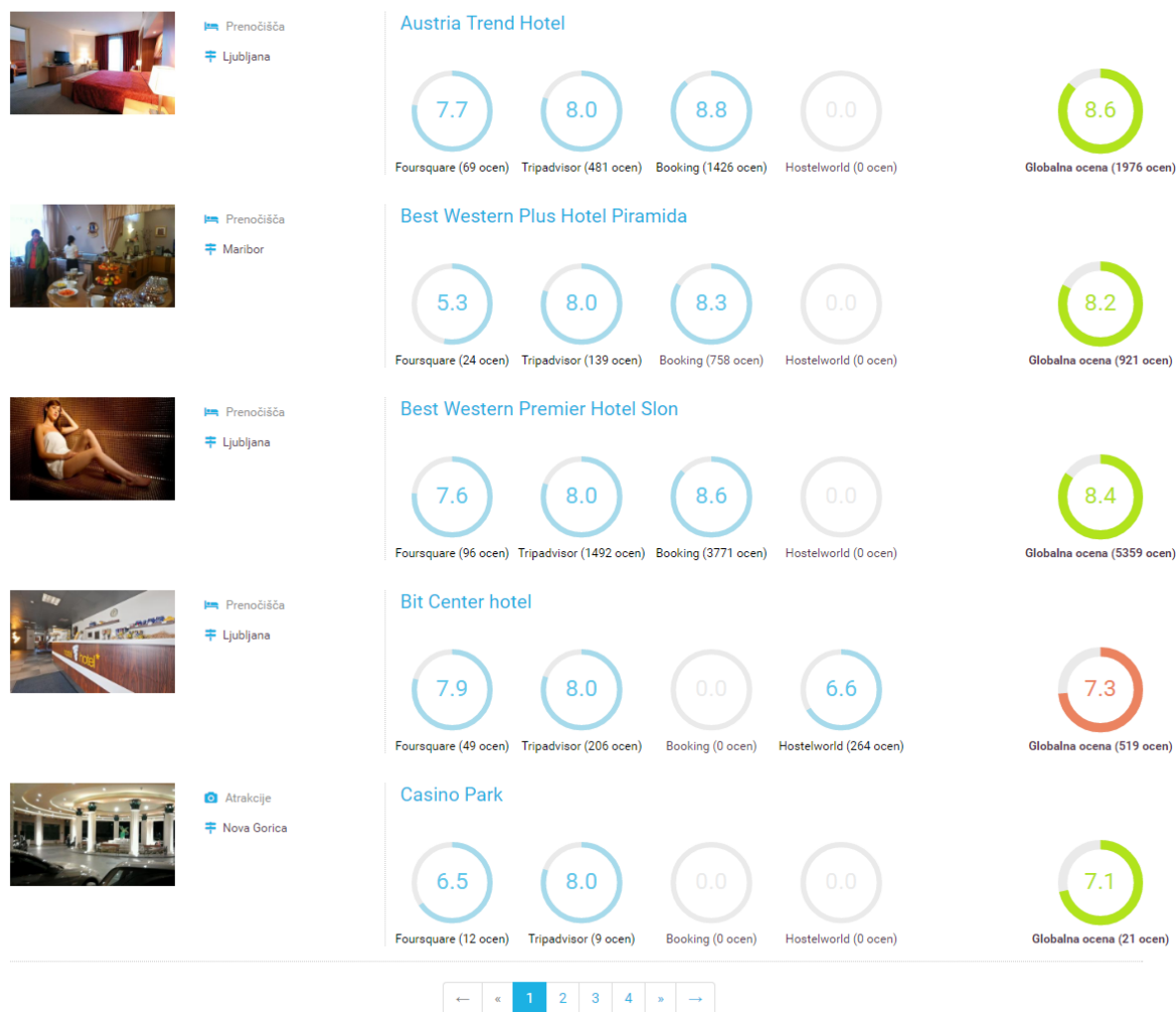
The image shows a search interface titled "Iskalnik turističnih točk". It contains three input fields: "Vse kategorije" with a dropdown arrow, "Vsi kraji" with a dropdown arrow, and "Turistična točka". To the right of these fields is a blue button labeled "Išči".

Slika 5.12: Iskalnik turističnih točk.

Število prikazov turističnih točk na seznamu je omejeno na število, ki je določeno v nastavitvah nadzorne plošče (slika 5.10). Za prikaz večjega števila turističnih točk je na voljo paginacija, ki je v nogi seznama, preko katere se lahko premikamo po celotnem seznamu najdenih turističnih točk.

Vsaka turistična točka ima prikazano sliko, kategorijo, kraj nahajanja ter ocene tekočega dneva. Te so dodatno porazdeljene v krogih za vsak spletni vir posebej in vsebujejo povprečno oceno ter število vseh ocen. Na koncu sledi še globalna ocena (poglavje 5.1.5), ki je obarvana glede na mesečni trend turistične točke ter seštevek števila vseh ocen. Ob kliku na sliko ali ime turistične točke se nam prikaže podstran (slika 5.14), ki vsebuje njene podrobnejše podatke.

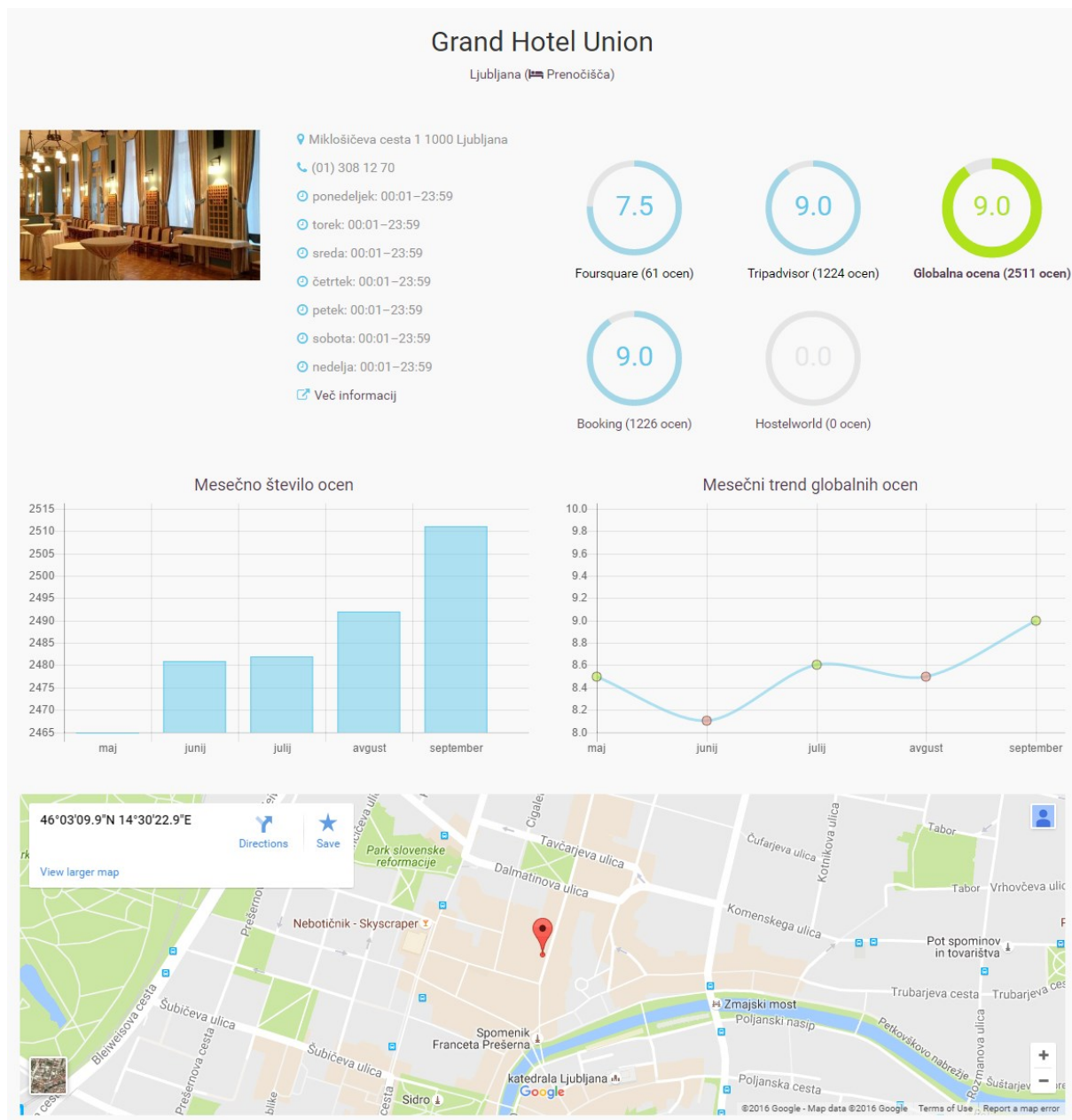
## Seznam najdenih turističnih točk

Najdenih je bilo **100** turističnih točk

Slika 5.13: Seznam najdenih turističnih točk (vrednosti so za nazornejši primer prilagojene).

Podstran s podrobnostmi turistične točke je sestavljena iz več elementov, ki prikazujejo različne podatke. Najprej je prikazanih nekaj splošnih podatkov, kot so ime turistične točke, kraj in kategorija. Nato sledi prikaz dodatnih informacij, kot so naslov, telefon, odpiralni čas in povezava na domačo stran. Ponovno imamo prikazane vse ocene spletnih virov, kot smo jih imeli že na seznamu najdenih turističnih točk. Poleg ocen pa imamo prikazana še 2 grafa, ki služita prikazu ocen po mesecih. Prvi prikazuje seštevek števila ocen turistične točke po mesecih, drugi graf pa prikazuje mesečni trend globalnih ocen. Za lažjo orientacijo imamo na voljo še zemljevid, na katerem je označena lokacija turistične točke.





Slika 5.14: Podrobnejši podatki o turistični točki (vrednosti so za nazornejši primer prilagojene).

### 5.3.2.2 Kraji

Na podstrani kraji imamo podoben iskalnik (slika 5.15) kot za turistične točke, le da imamo tokrat na voljo le iskanje po imenu kraja. Ob kliku na gumb »išči« se izvede iskanje kraja in se prikaže seznam najdenih krajev (slika 5.16) in število le-teh.

Slika 5.15: Iskalnik krajev.

Seznam najdenih krajev je sestavljen le iz imena kraja in njegovih ocen. V krogih so tokrat ocene prikazane kot povprečje seštevka ocen vseh turističnih točk izbranega kraja za vsak spletni vir posebej. Pod vsakim krogom pa je prikazan še seštevke ocen izbranega vira. Globalna ocena (poglavje 5.1.5) se tudi pri krajih izračuna na podoben način kot pri turističnih točkah. Ob kliku na naslov se nam odpre podstran (slika 5.17) s podrobnejšimi podatki o kraju.

#### Seznam najdenih krajev

Najdenih je bilo **5** krajev

##### Ljubljana

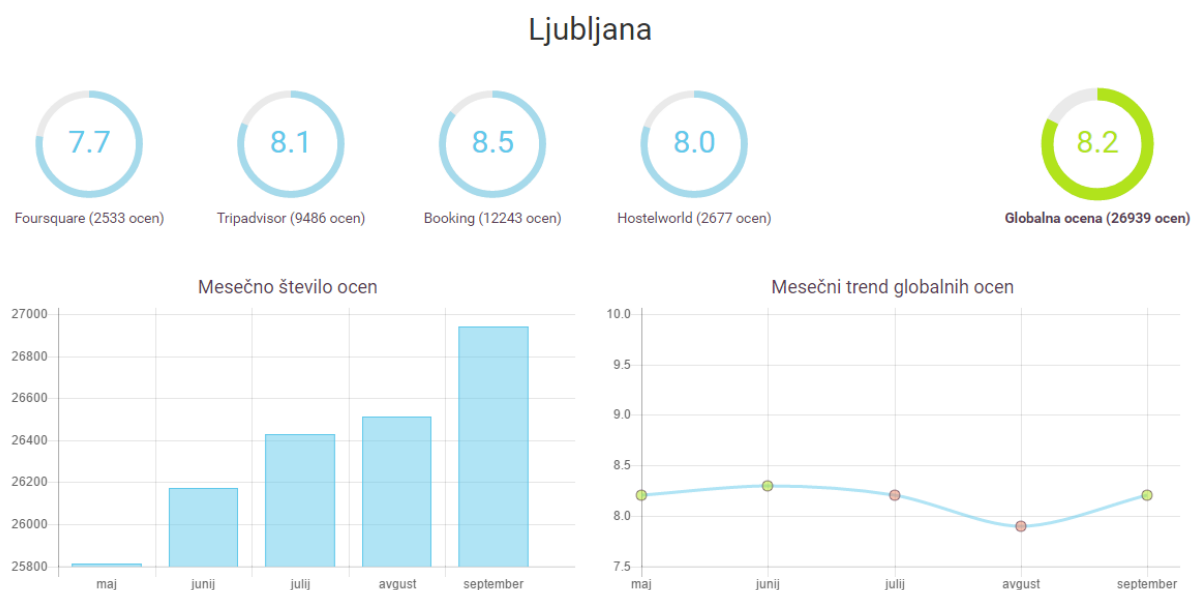


##### Maribor



Slika 5.16: Seznam najdenih krajev.

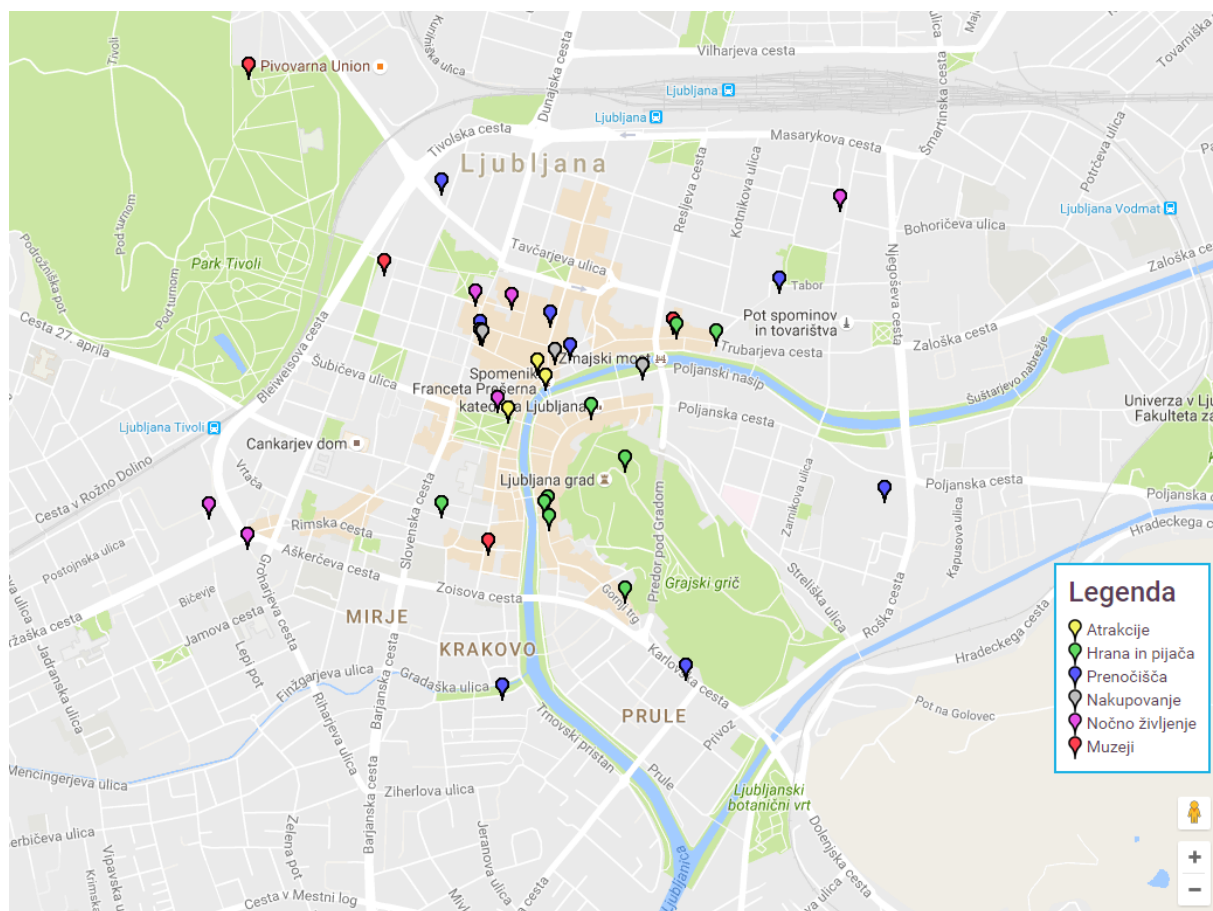
Tudi podstran, ki prikazuje podrobnosti izbranega kraja, je sestavljena na isti princip kot za turistično točko, le da vsebuje manj elementov in podatkov. Poleg imena kraja so prikazane ocene posameznih spletnih virov, kot na seznamu najdenih krajev. Prav tako kot pri posamezni turistični točki sta tudi pri kraju prikazana še 2 grafa. Prvi prikazuje skupno mesečno število ocen turističnih točk kraja. Drugi graf pa prikazuje skupni mesečni trend globalnih ocen turističnih točk kraja.



Slika 5.17: Podrobnejši podatki o kraju (vrednosti so za nazornejši primer prilagojene).

### 5.3.2.3 Zemljevid

Poleg možnosti iskanja turističnih točk preko iskalnika imamo na voljo v osrednji aplikaciji tudi možnost iskanja turističnih točk preko zemljevida (slika 5.18). Ta vsebuje obarvane točke po kategorijah z vključeno legendo na desni strani. Ob kliku na izbrano obarvano točko se nam prikažeta ime turistične točke in možnost klika na podrobnejši pregled, ki nas preusmeri na obstoječo podstran turistične točke.



Slika 5.18: Zemljevid turističnih točk.

## 5.4 Samodejno posodabljanje ocen – sistemski klic Cron

Glede na naše potrebe po vsakodnevnem pridobivanju ocen turističnih točk in omejitvah, ki jih imamo pri spletnem luščenju in uporabi programskih vmesnikov, smo se odločili, da bomo uporabljali sistemski klic Cron (ang. cron job) za samodejno pridobivanje ocen, saj je najbolj primerna rešitev za samodejno izvajanje ponavljajočih se nalog. Uporablja se na sistemih Unix preko ukazov ali skript in običajno avtomatizira določene potrebe, ki temeljijo na določenem času ali intervalu [18]. V našem primeru ga uporabljamo preko spletnega orodja Cpanel, ki ponuja možnost uporabe tega systemskega klica.

Ukaz je sestavljen iz najmanj šestih osnovnih atributov:

1. minute (vrednosti od 0 do 59),
2. ure (vrednosti od 0 do 24),
3. dan v mesecu (vrednosti od 1 do 31),

4. mesec (vrednosti od 1 do 12),
5. dan v tednu (vrednosti od 0 do 6, število 0 predstavlja nedeljo),
6. ukaz za izvršitev.

Nekaj možnih primerov Cron vrednosti z opisom je predstavljenih v tabeli (tabela 5.1).

Tabela 5.1: Primeri možnih časov Cron vrednosti z opisom.

Cron vrednost	Opis vrednosti
<b>0 * * * *</b>	Vsako uro na začetku ure
<b>15 * * * *</b>	Vsaki 15 min
<b>0 0 * * *</b>	Enkrat na dan ob polnoči
<b>30 08 10 06 *</b>	Vsako leto 10. 6. ob 8:30

V našem primeru se je moral ukaz izvesti vsak dan ob polnoči in sproži zagon PHP-datoteke, ki vsebuje skripto za pridobivanje ocen turističnih točk in krajev. Za dodatno varnost smo določili še dodaten parameter »hash« z naključnim nizom, ki se mora ujemati z nizom, ki je določen v PHP-skripti, preden se lahko izvede celotna programska koda (programska koda 5.11).

```
0 1 * * * /usr/bin/php5 domena.com/cronjob.php hash=wAAZzn32h1lhXOGzaZs1
>/dev/null 2>&1
```

Programska koda 5.11: Cron sistemski klic.

Ko se programska koda skripte izvede in zaključi dobijo določeni uporabniki e-obvestilo (slika 5.19), ki vsebuje število posodobljenih ocen turističnih točk in krajev, datum posodobitve in koliko časa je bilo potrebnega za posodobitev.

Ocene lokacij (100) in krajev (5) so bile uspešno posodobljene na datum 6.9.2016.  
Posodobitev ocen je potrebovala 00:03:51 časa.

Slika 5.19: E-obvestilo o posodobitvi ocen.



## **Poglavje 6      Sklepne ugotovitve**

V diplomskem delu smo razvili prototip sistema odzivne spletne aplikacije za iskanje in pregled ocen turističnih točk, ki se pridobivajo iz več spletnih virov ter nadzorno ploščo, ki služi upravljanju turističnih točk in krajev. Cilj diplomskega dela je bil razviti primerno in uporabniku všečno spletno aplikacijo, na kateri lahko na preprost in uporaben način išče turistične točke in kraje ter pridobi njihove podrobnejše ocene in ostale informacije. Treba pa je bilo tudi zagotoviti preprosto in pregledno nadzorno ploščo, preko katere lahko uporabniki sistema upravljajo turistične točke in kraje. Ker je treba ročno vnašati podatke, je bilo najbolj pomembno, da je to omogočeno na hiter in nazoren način. Vsebina diplomskega dela je predstavljena po fazah celotnega poteka razvoja prototipa, tehnologij in metod, ki so bile uporabljene pri razvoju.

Razvoj celotne spletne aplikacije je potekal brez večjih težav, saj je na spletu ogromno dokumentacije in primerov za razvoj takšnih aplikacij. Manjše težave so se pojavljale le pri spletnem luščenju podatkov, saj je bilo treba razviti rešitev, ki je omogočila dovolj hitro pridobivanje podatkov.

Delovanje prototipa spletne aplikacije, naložene na testno okolje, se je izkazalo za zanesljivo in uspešno. Največje pomisleke smo imeli, da bodo nastale težave z vsakodnevnim samodejnim pridobivanjem ocen, ki je najbolj kritična točka sistema, saj bi bilo mogoče, da se v primeru napake Cron skripta ne bi izvedla, zato bi morali to skripto naknadno ročno pognati, da bi pridobili ocene tega dneva. Takšna napaka bi lahko nastala v primeru, da se spremeni HTML-struktura spletnega vira in bi bila potrebna sprememba programske kode za spletno luščenje in s tem pridobivanje ocen. Na to težavo nismo naleteli, bi pa bilo kljub temu treba postaviti še dodatne varnostne pogoje v programski kodi v primeru, da se HTML-struktura spremeni, da se pridobivanje ocen iz te spletne strani začasno ne izvede. Nekaj manjših težav je bilo pri neskladnosti prikazovanja podatkov o ocenah, ki pa smo jih po testiranju odpravili.

Prototip sistema bi bilo mogoče nadgrajevati v več smereh. Najbolj pomembna bi bila dodatna avtomatizacija vnašanja turističnih točk v nadzorni plošči na takšen način, da bi URL-povezave za pridobivanje ocen samodejno pridobivali glede na vneseno ime in naslov izbrane turistične točke. Ena izmed možnih rešitev je, da bi glede na vneseno ime pridobili

najprej naslov turistične točke. Nato pa bi preko imena in naslova s spletnim luščenjem iskali te turistične točke za vsak spletni vir posebej. Na takšen način bi prišli do pravilnejših podatkov kot pri iskanju le po imenu, saj imajo lahko turistične točke enako oz. podobno ime. Zaradi zahtevnosti iskanja in združevanja takšnih podatkov preko spletnega luščenja ta funkcionalnost ni bila razvita, saj bi razvoj presegal obseg diplomskega dela. Naslednji večji korak pa bi bila izdelava domorodne mobilne aplikacije z možnostjo prikaza turističnih točk in ocen na podlagi naše trenutne lokacije. Zanimiva dodatna funkcionalnost bi bila tudi omogočanje izvoza podatkov ocen, ki bi se jih lahko naknadno uporabljalo za različne analitične potrebe. Smiselno bi bil tudi razvoj nekaj drugih manjših funkcionalnosti, kot so prikaz podrobnejše statistike ocen turističnih točk in krajev skozi čas, prikaz dodatnih informacij o turističnih točkah z možnostjo pregleda komentarjev ocen in izboljšanje iskalnika z možnostjo večje izbire parametrov pri iskanju ter sortiranju prikazanih rezultatov.



## Literatura

- [1] B. Brinzarea, C. Darie in A. Hendrix, AJAX and PHP: Building Modern Web Applications, 2nd Edition, Packt Publishing, 2009.
- [2] M. Delisle, Mastering phpMyAdmin 3.4 for Effective MySQL Management, Packt Publishing, 2012.
- [3] D. S. McFarland, JavaScript & jQuery: The Missing Manual, 3rd Edition, O'Reilly Media, 2014.
- [4] S. F. Rahman, Jump Start Bootstrap, SitePoint, 2014.
- [5] D. Sklar, Learning PHP: A Gentle Introduction to the Web's Most Popular Language, O'Reilly Media, 2016.
- [6] M. Turland, Php|architect's Guide to Web Scraping, musketeers.me, LLC, 2010.
- [7] J. Ward, Instant PHP Web Scraping, Packt Publishing, 2013.
- [8] Sublime Text [Online]. Dosegljivo: [https://en.wikipedia.org/wiki/Sublime\\_Text](https://en.wikipedia.org/wiki/Sublime_Text). [Dostop avgust 2016].
- [9] PhpMyAdmin [Online]. Dosegljivo: <https://www.phpmyadmin.net/>. [Dostop avgust 2016].
- [10] What is cPanel [Online]. Dosegljivo: <https://www.greengeeks.com/kb/3708/what-is-cpanel>. [Dostop avgust 2016].
- [11] Why learn PHP [Online]. Dosegljivo: <http://www.bestprogramminglanguagefor.me/why-learn-php>. [Dostop avgust 2016].
- [12] MySQL introduction [Online]. Dosegljivo: <http://www.tutorialspoint.com/mysql/mysql-introduction.htm>. [Dostop avgust 2016].

- [13] Why learn HTML5 [Online]. Dosegljivo: <http://www.bestprogramminglanguagefor.me/why-learn-html5>. [Dostop avgust 2016].
- [14] JSON [Online]. Dosegljivo: <https://en.wikipedia.org/wiki/JSON>. [Dostop avgust 2016].
- [15] What is JSON [Online]. Dosegljivo: <http://www.makeuseof.com/tag/json-laymans-overview/>. [Dostop avgust 2016].
- [16] 10 Best reasons to use Bootstrap for amazing web designs [Online]. Dosegljivo: <https://www.devsaran.com/blog/10-best-reasons-use-bootstrap-amazing-web-designs>. [Dostop avgust 2016].
- [17] A Practical Guide to Web Scraping [Online]. Dosegljivo: <https://leanpub.com/web-scraping>. [Dostop avgust 2016].
- [18] Cron [Online]. Dosegljivo: <https://en.wikipedia.org/wiki/Cron>. [Dostop avgust 2016].